Издательство «Знание»

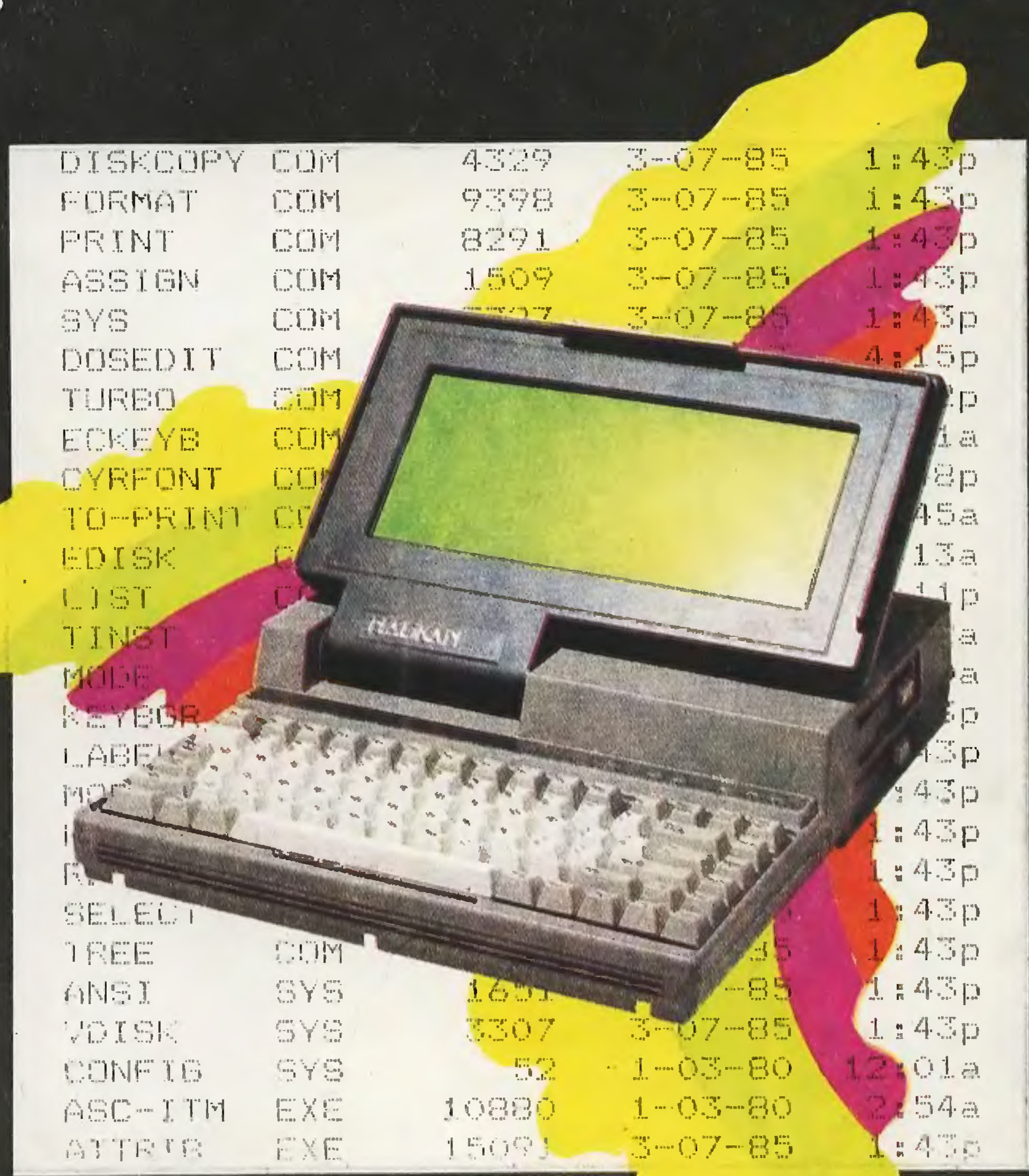
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ЕЁ ПРИМЕНЕНИЕ

Новое в жизни, науке, технике

Подписная научно-популярная серия

Издается ежемесячно с 1988 г.

Операционные системы



1990

1

Новое
в жизни,
науке,
технике

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ЕЁ ПРИМЕНЕНИЕ

Подписная
научно-
популярная
серия

1/1990

Издается
ежемесячно
с 1988 г.

ОПЕРАЦИОННЫЕ СИСТЕМЫ

В номере:

В номере:
В. Н. ЧЕТВЕРИКОВ
ИЗ ЖИЗНИ ОПЕРАЦИОННЫХ СИСТЕМ 3

В. Я. ЦВЕТКОВ
ОПЕРАЦИОННАЯ СИСТЕМА MC—DOC 20

ОПЕРАЦИОННАЯ СИСТЕМА CP/M 32

РУБРИКИ
Языки программирования
ЭВМ.Микропроцессоры.Робототехника
Переводы
Персоналии
Мы и ПАПА

Издательство
«Знание»
Москва
1990

ББК 32.973
О-60

Авторы выпуска

ЧЕТВЕРИКОВ Василий Николаевич — кандидат технических наук, доцент кафедры вычислительной техники Московского горного института, специалист в области управления информационными процессами.

ЦВЕТКОВ Виктор Яковлевич — кандидат технических наук, автор 133 научных работ, среди них 9 изобретений, один патент ГДР.

МАЛЫХИНА Мария Петровна — кандидат технических наук, доцент, программист.

ЧАСТИКОВ Аркадий Петрович — кандидат технических наук, доцент, специализируется в области информатики и вычислительной техники.

УТЕНКОВ Сергей Альбертович — инженер, журналист.

ТАРАСЕНКО Борис Алексеевич — старший научный сотрудник НПО «Энергия».

КОЛОСОВ Юрий Владимирович — кандидат технических наук, доцент, специалист в области разработки средств микропроцессорной техники.

РЕДАКТОР Б. М. ВАСИЛЬЕВ

ISBN 5-07-000336-4

© Издательство «Знание», 1990 г.

Дать точное и краткое определение операционной системы затруднительно, если вообще возможно. Наиболее часто встречается в литературе определение: ОС — это важнейшая часть программного оснащения компьютера, представляющая собой комплекс управляющих и обслуживающих программ, обеспечивающих максимальную эффективность работы вычислительной системы за счет автоматического управления вычислительными процессами и ресурсами.

В. Н. ЧЕТВЕРИКОВ

ИЗ ЖИЗНИ ОПЕРАЦИОННЫХ СИСТЕМ

Можно ли прожить без ОС

История операционных систем — это цепь попыток удовлетворения двух иногда конфликтующих и взаимодополняющих потребностей. Одна из них заключается в необходимости управления материальными (аппаратура) и программными ресурсами вычислительной системы с целью обеспечения ее эффективной и правильной работы. Другая состоит в предоставлении пользователям различных служб, которые преобразуют совокупность компонент вычислительной машины в некоторую виртуальную установку, гораздо более пригодную для использования в прикладных системах. Операционные системы адаптировались и изменялись в течение ряда лет в соответствии со многими изменениями в количестве, качестве и существе ресурсов, которыми они должны управлять, и услуг, которые они должны оказывать.

Ранние операционные системы, интенсивно использовавшиеся в середине 60-х годов, управляли множеством аппаратных ресурсов, которые значительно отличались по своей емкости, быстродействию и степени связности от современного оборудования. Среди новых обязанностей операционных систем, появившихся в связи с изменениями в аппаратуре, можно указать управление средствами двумерного вывода данных (дисплеи и лазерные принтеры), управление и планирование работы нескольких параллельно действующих процессоров, обеспечение связи между вычислительными установками, образующими распределенную систему. Классы программных ресурсов и служб, реализуемых и управляемых операционной системой, в основном соответствуют классам аппаратных ресурсов и могут быть разделены на три группы: службы организации связи, которые покрывают все формы ввода-вывода; общения ЭВМ с внешним для нее миром; службы хранения и обработки данных.

То, чего всегда не хватает

Познакомимся поближе с основными ресурсами современной вычислительной машины, взяв в качестве примера персональную ЭВМ фирмы IBM.

Процессор. Центром вычислительной системы является ее процессор. Именно процессор выполняет команды, входящие в состав программы для ЭВМ. В персональной ЭВМ используется микропроцессор 8088 или его «брат» 8086. Оба процессора являются 16-разрядными, т. е. могут выполнять операции над 16 двоичными разрядами одновременно, но микропроцессор 8088 может обмениваться с другими узлами ЭВМ только 8-разрядными данными, а процессор 8086 — 16-разрядными.

Процессор управляет основными операциями ЭВМ посредством передачи и приема управляющих сигналов, адресов памяти и данных от одной части ЭВМ к другой через совокупность линий связи, называемых шиной. Вдоль шины расположены входные и выходные порты, которые соединяют с ней различные микросхемы электронного обрамления. Данные проходят через эти порты при движении от процессора к другим частям ЭВМ или наоборот.

Внутри процессора 8088 находятся 14 регистров, образующих рабочую область для передачи и обработки данных. Эти внутренние регистры обеспечивают временное хранение данных, адресов памяти, указатели на команды и флагов состояния и управления. С их помощью процессор может получить доступ к памяти объемом более 1 млн. байт и ко всем (потенциально свыше 64 тыс.) портам ввода-вывода.

ЭВМ должна быть способна реагировать на события, происходящие за пре-

делами ее микропроцессора, — например, на нажатие клавиши на клавиатуре. Имеются два способа, как это можно было бы сделать. Первый заключается в постоянном опросе всех частей ЭВМ: не нужно ли где-либо, что-либо предпринять. Разумеется, такой метод работы достаточно расточителен — он требует очень много процессорного времени. Другой метод заключается в том, чтобы позволить процессору спокойно выполнять ту или иную работу до тех пор, пока что-либо извне не захватит его внимание. Такой подход получил название работы с прерываниями. Использование прерываний является мощным фактором организации эффективной работы ЭВМ.

Механизм прерываний устроен следующим образом. Каждому виду прерываний сопоставлен номер. В памяти ЭВМ хранится таблица, определяющая программы, которые должны быть активизированы при возникновении прерывания того или иного вида. Таким образом, при возникновении прерывания «X» значение указателя следующей команды текущей программы запоминается и заменяется на значение, определяющее первую команду программы обработки прерывания «X». Когда программа обработки прерывания завершает свою работу, она с помощью специальной команды восстанавливает значение указателя следующей команды и тем самым возвращается управление исходной программе.

Понятие прерывания было создано как практический способ организации взаимодействия процессора с внешним миром. Но очень скоро стало понятно, что это понятие применимо и для внутренних нужд. Поэтому наряду с прерываниями от аппаратуры имеются так называемые логические и программные прерывания.

Логические прерывания генерируются самим микропроцессором 8088, когда он встречает некоторые необычные ситуации. Например, прерывание 0 генерируется, когда обнаруживается попытка деления на 0.

Программные прерывания являются наиболее интересным типом прерываний. Когда программа нуждается в использовании другой программы или подпрограммы, она должна передать управление процессором этой программе. Прерывание позволяет это сделать и является альтернативой более традиционному способу, связанному с использованием команды CALL. Имеются две причины, объясняющие, почему прерывание используется вместо непосредственной адресации. Наиболее важно, что такой способ позволяет изменять вызываемую программу. Изменение может повлиять на размер и расположение программы в памяти. Но поскольку эта программа вызывается прерыванием, то нет необходимости в изменении вызывающей программы. Другая причина использования программных прерываний состоит в возможности замены программы обработки прерываний. Для этого необходимо изменить лишь соответствующий элемент таблицы прерываний.

Возможности процессора 8088 в части арифметических операций ограничены целочисленной арифметикой. Операции над вещественными числами, с плавающей запятой, могут быть выполнены одним из двух способов. Первый и наиболее общий способ состоит в программировании — написании подпрограмм, которые, используя логические операции и операции целочисленной арифметики, реализуют операции над вещественными числами. Другой путь связан с использованием специализированного сопроцессора.

Микропроцессор 8088 сконструирован так, что позволяет включить в вычислительную систему еще один процессор — сопроцессор 8087. Этот процессор имеет достаточно узкую специализацию: быстрая обработка вещественных чисел. Кроме обычных операций: сложение, вычитание, умножение и деление, он может выполнять и более экзотические операции, например вычисление тригонометрических функций. Специальные сигналы, предусмотренные в конструкции процессора 8088, позволяют ему передать задание процессору 8087 и получить от него результаты.

Оперативная память. Другим немаловажным ресурсом вычислительной системы является память. Внутренняя

или оперативная память ЭВМ — это то место, где хранится информация, обрабатываемая процессором. Здесь же хранятся и выполняемые программы. Память организована как совокупность ячеек, каждая из которых вмещает один байт или 8 бит. Все ячейки пронумерованы, начиная от нуля. Номер ячейки называется адресом и используется для обращения к ней. Поскольку адрес — это число, то над ним допустимы арифметические операции.

Конструкция любой ЭВМ накладывает ограничения на максимальное значение адреса и тем самым на размер памяти. В процессоре 8088 адрес имеет длину 20 разрядов, и поэтому процессор располагает адресным пространством, равным двум в двадцатой степени, или свыше 1 млн. байт.

Арифметические возможности процессора 8088 ограничены работой с 16-разрядными числами. Поэтому необходимы практические методы для работы с 20-разрядными адресами. Решение этой проблемы связано с использованием двух 16-разрядных чисел. Одно из чисел рассматривается как дополненное справа четырьмя нулевыми разрядами. И следовательно, имеющее длину 20 бит. Это число называется сегментной частью адреса. Второе число не сдвигается, а используется так, как оно есть, и называется относительной частью адреса или смещением, поскольку рассматривается как номер ячейки, но не с начала памяти, а от ячейки, определяемой сегментной частью адреса. Сумма этих двух чисел (при сдвинутом на четыре разряда первом) и определяет адрес ячейки.

Для использования описанной сегментной адресации в микропроцессоре 8088 выделены специальные регистры, предназначенные для хранения сегментной части адреса. Загрузив сегментный регистр, мы получаем доступ к 64К адресам, следующим за адресом сегмента. Чтобы иметь возможность работы с более чем 64К байт памяти, микропроцессор 8088 имеет четыре сегментных регистра — каждый со специальным назначением, связанным с использованием памяти. Поскольку память используется для хранения обрабатываемых данных и результатов, то два из регистров предназначены для указания на программу и на данные. Третий используется для организации стека, а четвертый — для расширения области хранения данных.

Имея свыше 1 млн. байт, можно зарезервировать часть адресного пространства для специальных целей. Поэтому логически единое пространство памяти физически разделено. Часть памяти (64К) обычно размещается вместе с процессором на системной плате, именно в ней размещается таблица векторов прерываний (в младших 1024 байтах). Другая часть оперативной памяти (до 576К) находится на плате расширения.

Адреса от A0000 до DFFFF относятся к памяти, используемой для управления дисплеем, и соответствующее запоминающее устройство размещается на плате адаптера дисплея. Остальные адреса соответствуют ячейкам постоянного запоминающего устройства (ПЗУ), т. е. устройства, запись в которое проводится специальными средствами, и во время работы ЭВМ невозможна. Преимуществом такого устройства является постоянная готовность информации — она не стирается при выключении питания и не может быть использована сразу, как только питание будет включено.

Внешняя память. Оперативная память — это лишь временное хранилище данных. Отключение питания стирает все ее содержимое. Для долговременного хранения в персональной ЭВМ используются магнитные диски.

Гибкие диски или дискеты представляют собой круглые пластмассовые диски с нанесенным на них магнитным слоем. Такой диск заключен в предохраняющий его от повреждений конверт с прорезями, необходимыми для подвода головки считывания/записи и для обеспечения вращения диска. Информация записывается на диск по концентрическим окружностям, называемым дорожками. Объем дискеты зависит от количества дорожек на одной стороне дискеты, от возможности записи на обеих сторонах и плотности записи отдельных разрядов в рамках одной дорожки. Отдельные разряды обычно группируются на дорож-

ках в сектора размером 512 байт. Число секторов на каждой дорожке одно и то же, несмотря на уменьшение их длины по мере уменьшения радиуса, и может быть равно 8; 9 или 15. Принимая во внимание возможность записи с двух сторон дискеты и способность накопителей на магнитных дисках вести запись-чтение с 40 или 80 дорожек на сторону, можно вычислить, что емкость дискеты в зависимости от формата записи составляет 160, 180, 320, 360, 720 Кбайт или 1.2 Мбайт.

Гораздо большей емкостью обладают жесткие диски, также включаемые в конфигурацию некоторых моделей персональных ЭВМ. Жесткие диски (диски типа «Винчестер») изготавливаются из алюминия, они несъемные, поскольку жестко закреплены в накопителе. Кроме этого, благодаря поддержанию специальной атмосферы внутри накопителя удалось значительно увеличить скорость вращения диска и уменьшить расстояние между диском и считывающей головкой. Соответственно сократилось время, затрачиваемое на чтение информации с диска, и увеличилась его емкость без существенного увеличения габаритов. Емкость накопителя на жестких дисках зависит от числа пластин и головок и может варьироваться в различных устройствах от 5 до 100 Мбайт и выше.

Дисплеи. Дисплей позволяет перевести информацию в вид, воспринимаемый человеком. Для подключения дисплея к ЭВМ необходимо специальное устройство — адаптер дисплея, основу которого составляет контроллер электронно-лучевой трубки. Существует несколько видов адаптеров, но все они так или иначе моделируют два первоначально реализованных фирмой IBM адаптера: монохромный и цветной графический адаптер.

Все режимы, в которых могут работать дисплеи, разделяются на текстовые и графические. Текстовые режимы позволяют выводить только символы, а в графическом режиме можно получить как сложные рисунки, так и символы различного вида и размера.

Цветной графический адаптер может работать и в текстовом и в графическом режимах. Он, как и монохромный адаптер, использует для управления дисплеем метод отображения памяти. Суть метода в использовании участка памяти для представления изображения, которое должно быть выведено на экран дисплея. Аппаратура адаптера циклически читает информацию из памяти и отображает ее на экране.

Таким образом, для вывода информации на экран дисплея нужно лишь занести ее в соответствующие ячейки памяти, и при очередном просмотре этой памяти адаптер выводит нужное изображение на экран дисплея.

Изображение обновляется 60 раз в секунду. В конце каждого цикла обновления электронный пучок перемещается из правого нижнего угла экрана в левый верхний. Именно этот промежуток времени (1.25 мс) стараются использовать для обновления информации в памяти дисплейного адаптера. В противном случае при одновременной попытке изменения какого-либо байта со стороны процессора и чтения этого байта адаптером на экране появляются помехи в виде «снега».

Способ представления изображения в памяти зависит от режима. Всего имеется 16 режимов. Цветной графический адаптер обеспечивает поддержку лишь первых семи. Первые четыре (0, 1, 2, 3) — текстовые режимы и три (4, 5, 6) — графические.

В любом текстовом режиме, а отличаются они лишь числом строк на экране (40 или 80) и возможностью использования цвета, каждой позиции на экране соответствуют 2 байта памяти. Их содержимое определяет, что будет в данной позиции экрана и в каком виде. Первый байт интерпретируется как код символа, изображаемого на экране. Второй байт определяет атрибуты изображения: цвет или яркость и наличие подчеркивания.

Графические изображения образуются из отдельных точек. Разрешающая способность дисплея определяется числом строк или линий сканирования и числом точек в каждой строке. Число точек в строке одинаково для всех режимов — 200, а вот число строк различно: 320 для режимов 4,5 и 640 для режима 6. Для определения изображения необходимо задать характер

свечения каждой точки. Поэтому в режиме 6 должно использоваться по меньшей мере 128 000 бит или 16 Кбайт. А так как в режимах 4 и 5 используется в 2 раза меньше точек, то оставшаяся память может применяться для задания цвета точек, а не только их наличия или отсутствия. Таким образом, каждой точке соответствуют два разряда, которые могут содержать одно из четырех значений. Смысл этих значений определяется сложнее, чем простым соответствием код-цвет: значение 00 — любой из 16 цветов, остальные значения определяют цвет в зависимости от выбранной палитры. Палитр две. В первую палитру входят цвета зеленый (01), красный (10) и коричневый (11), а во вторую — синий, бордовый и белый.

Клавиатура. Внешне клавиатура выглядит весьма просто, но по своей сути она представляет собой целую ЭВМ со своим процессором, задача которой наблюдение за состоянием клавиш и передача сигналов о происшедшем основному процессору.

С точки зрения клавиатуры отдельные клавиши не имеют какого-либо смысла. Она лишь знает номер клавиши, называемый кодом сканирования. Когда нажимается какая-либо клавиша, клавиатура передает в основной процессор обычный код сканирования. Когда же клавиша опускается, то сообщается тот же код, но увеличенный на 128. Сигналы передаются через уже известные нам механизмы прерывания и порты ввода-вывода.

ОС персонально для вас

Так чем же занимается операционная система в рассматриваемом случае PC DOS.

По большей части она занята тем, что скрывает от нас кучу мелких и скучных деталей. Рассмотрим вкратце команду COPY в случае, когда нужно скопировать данные с одной дискеты на другую. О чем должна позаботиться операционная система.

— Есть ли копируемые данные на исходной дискете?

— Копируются ли данные на дискету или, например, на принтер?

— Нет ли копируемых данных на другой дискете?

— Достаточно ли пространства для создания копии?

— Есть ли место в каталоге?

— Какой формат имеет исходная дискета?

— Существуют ли указанные в команде накопители?

— Каков доступный размер промежуточного буфера в памяти?

Вы не устали? А ведь это только начало. Есть еще и физическая сторона проблемы.

— Включен ли двигатель накопителя?

— Подведена ли головка к нужной дорожке?

— Сколько секторов нужно прочитать с данной дорожки?

— Готов ли накопитель выполнить команду?

И далее.

Число различных аспектов управления вычислительной машиной очень велико, и справиться с созданием системы, способной учесть все необходимые вопросы, можно, лишь применив модульный принцип. Разработка такой сложной системы упрощается, если работа разделяется на четко выделенные части. Эти части объединяются в ясно определенную иерархию, где каждый уровень имеет дело с соответствующими деталями и скрывает их, делает их прозрачными для вышележащих уровней, пользуясь услугами нижележащих уровней.

В PC DOS можно выделить шесть составляющих. Первым, самым нижним является базовая система ввода-вывода (BIOS), записанная в постоянное запоминающее устройство. Фактически BIOS встроена в IBM PC и может быть использована как часть любой операционной системы для ЭВМ этого типа.

Следущая компонента DOS, не образующая уровень, — это программа начальной загрузки, очень короткая и простая программа, размещаемая в первом секторе каждой дискеты. Ее задача — начать процесс загрузки (записи в память) операционной системы при включении или рестарте ЭВМ.

Третья и четвертая компоненты хранятся на дискете в виде файлов IBMBIO.COM и IBMDOS.COM. Первый, IBMBIO.

COM, является расширением BIOS и может быть дополнен другими частями, называемыми независимыми драйверами устройств. Второй, IBMDOS.COM, обеспечивает поддержание файловой структуры, т. е. совокупности каталогов и другой справочной информации на диске. Оба эти файла относятся к «скрытым системным» файлам. Командой DIR они не включаются в список файлов, и их копирование производится особой командой.

Пятая компонента также хранится в файле, именуемом COMMAND.COM. Главная ее функция — обработка команд, передаваемых вами DOS. Внутренние команды выполняются подпрограммами, входящими в COMMAND. COM, а все внешние — отдельными программами, составляющими в совокупности шестую часть системы.

Разные компоненты DOS отличаются длительностью пребывания их в памяти ЭВМ. Если BIOS всегда присутствует в ЭВМ, то программа начальной загрузки помещается в память лишь на время загрузки. То же можно сказать о внешних командах, которые не являются резидентной частью DOS, поскольку не остаются в памяти на протяжении всей работы DOS. Резидентную часть DOS составляют два системных файла IBMBIO.COM и IBMDOS.COM вместе с драйверами устройств. COMMAND.COM образует отдельную категорию, поскольку является полурезидентной.

Компоненты операционной системы

BIOS. Программы, составляющие BIOS, обеспечивают наиболее примитивный и основополагающий сервис. Вся базовая система ввода-вывода размещается в ячейках памяти с адресами от FF000 до FFFFF (адреса шестнадцатиричные, соответствуют десятичным 1040384 и 1048575).

BIOS состоит из нескольких частей, большинство из которых являются программами. Первая выполняемая программа — это программа самопроверки по включению питания (POST — Power — On — Self — Testing). Эта программа выполняет проверку работоспособности памяти и оборудования, подключенного к ПЭВМ. Ее работа в основном и определяет задержку с загрузкой системы при включении питания.

Следующая выполняемая программа — программа запуска операционной системы. Эта программа проверяет наличие накопителя на гибком диске и пытается прочесть программу начальной загрузки. После чего она передает управление этой программе, чтобы она могла прочитать с дискеты остальные части операционной системы.

Кроме этих программ, в состав BIOS входит множество других, обеспечивающих доступ к возможностям стандартного периферийного оборудования. Для активизации этих сервисных программ используются программные прерывания: команда прерывания указывает на конкретный элемент таблицы векторов прерывания, который содержит адрес сервисной программы. Все услуги BIOS могли бы предоставляться через одно какое-то прерывание, но вместо этого они разделены на категории (по устройствам), и каждой категории соответствует свое прерывание. Такое решение увеличивает модульность конструкции, снижает объем изменений, связанных с заменой какой-либо из сервисных подпрограмм. Работа с BIOS обеспечивается через 12 прерываний, большая часть из которых позволяет вызывать несколько функций. Указание нужной функции осуществляется записью номера этой функции в регистр общего назначения AH.

Так, для работы с дискетой в BIOS включено шесть функций. Все они вызываются через прерывание 19.

Первая функция переводит накопитель на гибком диске в начальное состояние, что может потребоваться для восстановления его работоспособности после возникновения каких-либо проблем.

Вторая функция обеспечивает получение кода состояния накопителя, содержащего более подробную информацию об ошибке, чем просто ее индикацию. Например, этот код поможет распознать неверное указание номера сектора или ошибку чтения данных и т. д.

Третья и четвертая функции служат

для чтения и записи одного или нескольких секторов какой-либо дорожки. Следует, однако, отметить, что данные функции не выполняют проверку ошибочных состояний и автоматическое выполнение повторных попыток.

Пятая функция осуществляет проверку данных после операции чтения или записи. Она выполняет повторное чтение секторов дискеты с подсчетом избыточного циклического кода для выявления ошибок. Это именно та функция, что используется командой COPY при наличии ключа «/V».

Последняя, шестая, функция формирует дорожку дискеты, записывая адресную информацию для секторов и стандартное значение F6 (шестнадцатиричное) в область данных каждого сектора. В качестве параметров, определяющих работу функции, ей передаются номер дорожки, стороны дискеты, а также число секторов и код длины одного сектора.

Как мы выяснили ранее, управление дисплеем возможно посредством записи в соответствующие ячейки памяти. Но при такой работе с дисплеем у вас могут возникнуть трудности при смене дисплейного адаптера и, кроме того, часть возможностей, особенно связанная с управлением режимами и курсором, окажется недоступной.

BIOS представляет семнадцать функций для управления видеомонитором. Эти функции обеспечивают установку режима работы и определение текущего режима, установку и определение положения курсора на экране, вывод символа или отдельной точки вместе с определенными атрибутами и т. д.

Среди наиболее интересных следует отметить функции, обеспечивающие имитацию работы с телетайпом. В их число входят функции, обеспечивающие перемещение содержимого экрана вверх или вниз, наподобие движения бумаги в телетайпе. Они позволяют нам переместить в общем случае часть экрана вверх или вниз на несколько строк. Строки, выходящие за верхнюю (нижнюю) границу определенной части экрана (окна), исчезают, а на другом конце появляются пустые строки. Остальная часть экрана не изменяется. Таким образом, весь экран может быть разбит на отдельно управляемые части, определяемые координатами верхнего левого и нижнего правого углов. При этом для каждого окна может быть определен свой заполнитель пустой строки, что позволяет работать с различно окрашенными окнами, выделяющимися на общем фоне.

Имитацию вывода символа на телетайп выполняет другая функция. При этом она, следуя стандартным для телетайпа соглашениям, распознает четыре специальных кода и действует соответственно: например, код 7 производит звуковой сигнал. Коды, известные как «забой», «перевод строки» и «возврат каретки», обрабатываются соответственно.

Подпрограммы BIOS, доступные через прерывание 9, ответственны за определение кодов, соответствующих нажатию той или иной клавиши.

Часть задачи, решаемой BIOS, связана со слежением за изменениями возможных регистров клавиатуры. Следует отметить, что хотя в обычной пишущей машинке имеется два регистра: верхний и нижний, клавиатура многих ЭВМ типа IBM PC имеет три различных вида регистров.

Во-первых, нормальные действия клавиш для ввода строчных букв. Затем обычное переключение регистра для ввода прописных букв. Кроме этого, имеются два специальных регистра: дополнительный («Alt») и управляющий («Ctrl») регистры. Они действуют так же, как и обычный регистр, но как строчная «а» это не то же самое, что прописная «А», так и Ctrl-A и Alt-A — это в некотором смысле самостоятельные «символы».

Кроме регистровых клавиш, на клавиатуре IBM PC имеется четыре специальные клавиши — «переключатели», которые действуют как выключатели для некоторых определенных функций. Это клавиши: INSERT (вставка), CAPS-LOCK (прописные), NUM-LOCK (цифры) и SCROLL-LOCK (прокрутка). Две из них CAPS-LOCK и NUM-LOCK служат частью регистрового механизма клавиатуры, а две другие управляют спе-

циальными действиями.

Информация о состоянии этих четырех клавиш и о том, нажата ли клавиша временного переключения регистра, хранится BIOS в специально отведенном месте: в двух байтах с адресами 1047 и 1048. Содержимое этих байт отражает текущее состояние клавиатуры независимо от того, какая в этот момент выполняется программа.

Для обычных регистровых клавиш, которые активны только пока они нажаты, BIOS отмечает, нажаты они или нет, и соответственно вносит изменения в интерпретацию обычных клавиш. Именно для регистровых клавиш имеет особое значение дополнительный код сканирования, указывающий, что клавиша отпущена.

Когда BIOS получает код сканирования, соответствующий обычной клавише (не регистровой), она проверяет состояние регистров и затем транслирует код в код ASCII или специальный код клавиши.

Механизм кодирования, используемый BIOS для указания вводимого символа, применяет два байта. Если главный байт не равен нулю, вводимый символ соответствует одному из символов расширенного ASCII, и именно его код хранится в главном байте. В противном случае символ относится к специальным, и его код запоминается во вспомогательном байте. В число таких кодов входят, например, 40 кодов функциональных клавиш (10 обычных и 30 в ассоциации с регистровыми клавишами).

Некоторые комбинации клавиш BIOS обрабатывает специальным образом. Так, комбинация Ctrl—Alt—Del используется для того, чтобы потребовать перезагрузку операционной системы. Иногда, правда, это не происходит. Связана такая ситуация чаще всего с тем, что выполняющаяся в данный момент программа запретила прерывание процессора.

В виде каких же функций BIOS предстает перед вышележащим слоем в части управления клавиатурой? Таких функций всего три.

Первая возвращает следующий доступный символ. Управление она возвращает только после того, как символ будет получен. Поэтому эта функция не может использоваться для опроса готовности клавиатуры. Для этого предназначена вторая функция, которая опрашивает клавиатуру, не дожидаясь ввода очередного символа. Если клавиша к моменту опроса уже была нажата, то возвращается соответствующий код. Иначе отметка в регистре флагов обозначает, что символ еще не введен.

Наконец, последняя функция позволяет получить информацию о состоянии регистровых клавишей.

Кроме программ, в BIOS входят три таблицы, ссылки на которые также внесены в таблицу векторов прерываний.

Первая из этих таблиц содержит информацию, необходимую для инициализации видеомонитора: для каждого из режимов, отличающихся разрешающей способностью, хранится длина буфера дисплея, размер столбца и т. д.

Вторая таблица определяет параметры дискеты и накопителя на гибком диске. Наиболее важными параметрами являются временные характеристики: время перехода от одной дорожки к другой и время разгона двигателя и др., а также размер сектора в байтах.

Третья — это таблица генератора графики, которая используется для создания образов символов с кодами от 128 до 255 в любом из двух графических режимов.

Таким образом, BIOS скрывает от вышележащих слоев особенности периферийного оборудования. Поэтому замена оборудования может потребовать внесения изменений только в BIOS и значительно реже в вышележащие слои.

Программа начальной загрузки. Идея, положенная в основу работы программы начальной загрузки, используется почти в любой ЭВМ. Суть проблемы заключается в том, что работа ЭВМ может начаться только после загрузки программы в память машины, но сама загрузка уже требует работы ЭВМ. Поэтому осуществляется как бы раскрутка системы: сначала средствами BIOS считывается сектор диска, хранящий программу начальной загрузки, которая уже загружает основную

часть операционной системы. То есть в соответствии с английским названием этого процесса «bootstrap» система поднимает сама себя за шнурки ботинок.

Программа начальной загрузки размещается всегда в первом секторе нулевой дорожки дискеты. BIOS пытается считать этот сектор с дискеты, размещенной в накопителе А, и записывает программу начальной загрузки в стандартное место в памяти с адреса 31744. После этого управление передается считанной программе. Поскольку размер программы начальной загрузки ограничен 512 байтами, то она не может быть очень сложной, и для упрощения ее работы файлы IBMBIO.COM и IBMDOS.COM, которые она должна загрузить, размещаются в предопределенном месте на дискете. Благодаря этому не требуется организовывать поиск файлов, необходимый в общем случае. И именно поэтому форматирование «системных» дискет несколько отличается от форматирования обычных дискет, не содержащих систему, и обычные дискеты нельзя переделать в системные, поскольку там не зарезервировано место под два специальных системных файла.

IBMBIO.COM. Основная задача первого из системных файлов IBMBIO.COM состоит в расширении возможностей BIOS. С его помощью решаются три вопроса, не разрешимые в BIOS. Во-первых, IBMBIO.COM позволяет учесть специфические требования конкретной операционной системы PC DOS, поскольку BIOS обеспечивает лишь достаточно универсальные функции.

Во-вторых, с помощью IBMBIO.COM можно исправить ошибки, обнаруженные в BIOS. Конечно, такие программы, как BIOS, проходят весьма тщательную проверку, так как запись этих данных в ПЗУ делает их существенно неизменяемыми. Но если все же ошибки обнаружены или потребовалась замена программ, включенных в BIOS по другим причинам, то способ вызова программ в BIOS дает очень простой механизм замены их на программы, включенные в IBMBIO.COM. Достаточно лишь изменить соответствующую ссылку в таблице векторов прерываний. Так, например, изменяется таблица, определяющая параметры накопителя на гибких дисках, что позволяет существенно сократить время доступа к данным.

В-третьих, BIOS принципиально не может, а IBMBIO.COM обязан предоставлять возможность работы с новыми периферийными устройствами, которые могут быть подключены к IBM PC: графопостроители, мышь и другие. В ранних версиях PC DOS для этого требовалось модифицировать IBMBIO.COM. В настоящее время этого не требуется. Когда IBMBIO.COM начинает работу, он проверяет, нет ли на системной дискете файла конфигурации системы CONFIG.SYS. Если он находит такой файл, то читает из него команды, которые могут определять различные параметры системы, а также определять программы обслуживания устройств, драйверы устройств, которые должны быть включены в систему.

Файл IBMBIO.COM можно представить состоящим из драйверов стандартных устройств, которые называются резидентными драйверами в отличие от устанавливаемых или загружаемых, которые хранятся в отдельных файлах и необходимость загрузки которых указывается в файле конфигурации.

Все драйверы объединены в связный список. Заголовок каждого из них содержит указатель на следующий. Загружаемые драйверы всегда обрабатываются до резидентных. Поэтому для установки нового устройства CON (консольный дисплей) достаточно поименовать это устройство CON. Сканирование списка устройств прекращается при обнаружении первого соответствия. Таким образом, загружаемые драйверы имеют предпочтение.

Любой драйвер должен иметь специальный заголовок, расположенный в начале (с относительного адреса 0) программного сегмента, идентифицирующий его как драйвер устройства, определяющий точки входа и описывающий различные атрибуты устройства.

Выделены два вида драйверов устройств: символьные и блочные. Символьный драйвер обеспечивает после-

довательный ввод-вывод символов. Эти устройства имеют имя (например, CON, PRN, AUX и др.), и программы могут открыть каналы для выполнения операций ввода-вывода на эти устройства.

Блочные устройства — это дисководобные устройства. Они обеспечивают ввод-вывод информации, хранящейся в каком-либо одном секторе. Такой кусок информации обычно называется блоком. Такие устройства не имеют имени и поэтому не могут быть непосредственно открыты. Вместо этого они имеют номера устройств и идентифицируются буквами: A,B или C, например.

Один блочный драйвер может обслуживать одно или несколько устройств. Положение драйвера в списке всех драйверов определяет, каким устройством какие буквы будут соответствовать. Если первый в списке драйвер обслуживает четыре устройства, то они будут обозначаться буквами A, B, C и D. Если второй блочный драйвер обслуживает три устройства, то они получают обозначения E, F и G и так далее.

При обращении к драйверу ему передается адрес заголовка запроса. Заголовок запроса имеет фиксированную длину и формат, определяя длину передаваемых данных, требуемое устройство (для блочных драйверов) и код команды.

Драйверы устройств могут выполнять все или некоторые из фиксированного набора функций. Рассмотрим для примера действия блочного драйвера по команде записи.

1. Используя ссылку на заголовок запроса, драйвер читает код команды и определяет, что это запрос на запись.

2. Прочитав код устройства, драйвер определяет, на каком из накопителей должна быть выполнена запись.

3. Извлекает из пакета запроса адрес передаваемых данных, количество секторов и номер первого записываемого сектора.

4. Поскольку для указания сектора используется сквозная нумерация всех секторов дискеты, то драйвер переводит полученный номер сектора в номера стороны, дорожки и сектора на дорожке.

5. Выполняет запись данных, расположенных по адресу, указанному в запросе.

6. Завершив передачу данных, драйвер указывает реально переданное число секторов, успешность выполнения запроса и код возникшей ошибки, если таковая встретилась при выполнении запроса.

Таким образом IBMBIO.COM обеспечивает большую унификацию периферийных устройств, расширяя вместе с тем перечень устройств, которыми может управлять операционная система.

IBMDOS.COM. Второй из двух системных файлов файл IBM DOS.COM содержит совокупность подпрограмм, которые не только обеспечивают доступ к стандартным средствам ввода-вывода, но и предоставляют ряд услуг, связанных с управлением оперативной и дисковой памятью, организацией вычислительного процесса.

Особенность персональной ЭВМ — наличие только одного пользователя упрощает многие вопросы управления памятью и доступом к процессору, выдвигая на первый план вопросы управления дисковым пространством.

Большой объем дискового пространства, разнообразие данных, хранимых на дискетах и жестких дисках, послужили причиной создания и поддержания со стороны ОС некоторой информационной структуры на устройстве внешней памяти. Без такой структуры невозможно обеспечить организацию хранения и быстрого поиска данных.

Имеется много различных схем организации и хранения данных на дискете со своими преимуществами и недостатками. Основными принципами, определяющими стратегию PC DOS, в этом вопросе являются:

1. Все файлы данных хранятся в секторах стандартного размера по 512 байт.

2. Сектора выделяются из общего узла доступных секторов по мере необходимости без предварительного резервирования (исключение — файлы IBMBIO.COM и IBMDOS.COM).

3. Выделение секторов и логическая связь между секторами, образующими файл данных, обеспечиваются с

помощью таблицы распределения файлов (FAT).

4. Каждая дискета содержит корневой каталог или таблицу, в которой хранится перечень файлов, расположенных на этой дискете.

Таким образом, должны быть выделены четыре категории секторов, одна категория используется для хранения собственно данных, остальные три — для специальных целей.

Самый первый сектор выделяется для хранения программы начальной загрузки. Два следующих сектора занимает таблица распределения файлов. Наконец, последнюю категорию составляют сектора, выделенные под корневой каталог, семь секторов для двухсторонней дискеты.

Каждый элемент каталога, имеющий длину 32 байта, может содержать всю доступную информацию об одном файле или быть свободным. В элементе каталога выделено восемь полей.

Длина поля имени файла — 8 байт. Если имя файла имеет меньшую длину, то оно дополняется пробелами. Если значение первого байта поля имени файла 229, то это означает, что данный элемент каталога не используется. Когда файл стирается с дискеты, реально выполняется лишь следующее: освобождается занятое под файл пространство и в первый байт поля имени файла записывается 229. Поэтому вполне возможно в ряде случаев восстановить уничтоженный файл.

Поле расширения длиной 3 байта предназначено для хранения символов, определяющих тип файла.

Поле атрибутов длиной 1 байт определяет характеристики файла. Из 8 бит, составляющих байт, 2 не используются. Восьмой бит помечает файл как «доступный только для чтения». Помеченный таким образом файл не может быть изменен средствами DOS, для этого необходимо вначале снять эту пометку. Седьмой и шестой биты определяют файл как «скрытый» и «системный» соответственно. Установка любого из этих бит делает файл невидимым при обычном просмотре каталога. Пятый бит используется, чтобы пометить элемент каталога, содержащий метку дискеты. Метка занимает первые 11 бит и служит для идентификации дискеты. Четвертый бит указывает, что данный файл является каталогом. PC DOS, начиная с версии 2.0, поддерживает древовидную структуру каталогов, т. е. каждый каталог с корневого может содержать ссылку на подкаталог. Наличие подкаталогов снимает ограничение на число файлов, хранящихся на дискете, вытекающее из ограничения на размер корневого каталога (112 файлов для двухсторонней дискеты). Кроме того, подкаталоги служат прекрасным средством для логической организации файлов в группы.

Каждый подкаталог представляет собой файл, пространство под который выделяется обычным образом. Поэтому в отличие от корневого каталога подкаталоги могут расти в размере по мере необходимости.

Существуют три вида элемента каталогов, у которых установлен бит каталога. Если первый символ имени не является точкой (.), то элемент каталога определяет подкаталог данного каталога. Точка в качестве первого символа имени указывает, что данный элемент каталога используется для хранения информации о связях между каталогами. Каждый подкаталог имеет элемент с именем «.» и элемент с именем «..». Первый определяет ссылку на самого себя, а второй — ссылку на вышележащий (отцовский) каталог.

Следующий и последний используемый бит — третий. Он называется архивным битом и используется средствами сохранения и восстановления данных, хранящихся на фиксированном диске. Этот бит устанавливается при внесении изменений в файл. После сохранения файла на резервном устройстве бит сбрасывается.

В рамках элемента каталога выделено резервное поле размером 12 байт и два двухбайтовых поля, указывающих время и дату создания или изменения файла.

Двухбайтовое поле, называемое номер начального кластера, определяет начальную точку представления файла в FAT.

Последнее поле, поле размера файла, имеющее длину четыре байта, осо-

бенно важно для файлов, хранящих не текстовую информацию, так как является единственным способом определить точное местоположение конца данных, составляющих файл.

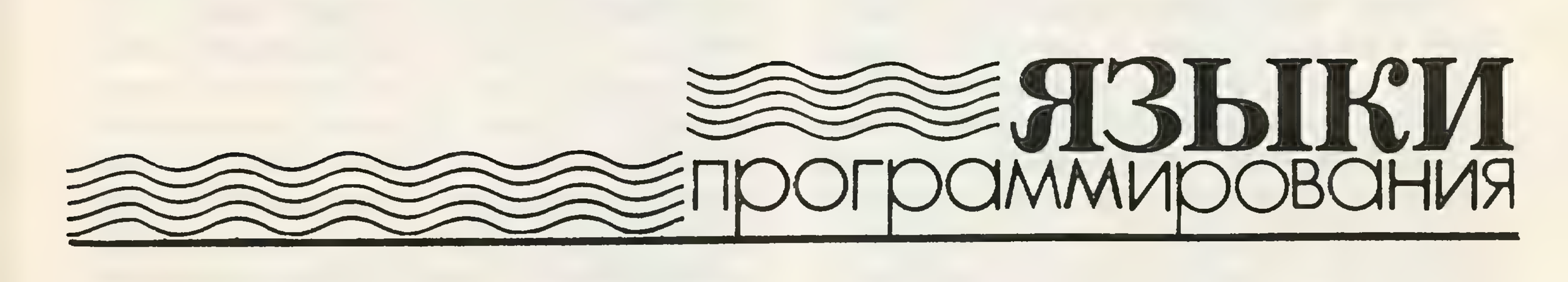
Выделение пространства дискеты управляется таблицей распределения файлов, основная идея которой состоит в том, что это простая таблица, где каждому кластеру, единице распределения дискового пространства, соответствует один элемент таблицы. Элементы таблицы могут указывать, что они используются или что они доступны для распределения. Доступные, свободные элементы указываются нулевым значением в таблице.

Кластеры, образующие пространство, отведенное конкретному файлу, соединены вместе, так, что они образуют «цепочку» — связный список. Элемент каталога, определяющий файл, содержит номер элемента FAT, соответствующего первому кластеру файла. В FAT каждый занятый элемент содержит номер следующего кластера файла, и так продолжается, пока не будет достигнут последний кластер файла. Соответствующий ему элемент FAT содержит маркер конца файла.

Размер кластера зависит от формата дискеты: для односторонних дискет он равен одному сектору, а для двухсторонних дискет — двум секторам. Это сделано для того, чтобы сохранить неизменным общее количество кластеров на дискете.

Таблица распределения файлов занимает один сектор, но для повышения надежности полностью дублируется, и поэтому под FAT фактически выделяются два сектора. Тем не менее FAT является наиболее уязвимым элементом файловой структуры, и потому многие программы-вирусы портят в первую очередь именно его.

Доступ к услугам, предоставляемым IBMDOS.COM, по-прежнему может



Русское название языка БЕЙСИК является транскрипцией английской аббревиатуры BASIC: Beginner's All — purpose Symbolic Instruction Code — универсальный код (язык) символических инструкций для начинающих.

История его создания связана с поисками путей решения проблемы: как обучать студентов доступному пониманию технике программирования, как научить их простому общению с ЭВМ.

Двое сотрудников Дартмутского колледжа Т. Куртц и Дж. Кемени занялись этой проблемой в конце 50-х годов. Прежде всего они поняли, что необходим такой компьютер, который был бы легок в изучении и использовании и не требовал больших усилий со стороны студентов. Причем было ясно, что студенты не захотят работать с перфокартами, поэтому для общения с ЭВМ необходим режим разделения времени и диалоговый режим работы с машиной.

Кроме этого, решению указанной проблемы содействовала необходимость разработки нового языка программирования, который легко можно было изучать и которым легко было бы пользоваться. Применение языка ассемблера или таких языков, как АЛГОЛ или ФОРТРАН, было вне обсуждения исходя из предшествующего опыта преподавания.

Летом 1963 г. началась работа над черновым вариантом БЕЙСИКа — нового языка. К этому времени группа под руководством Т. Куртца и Дж. Кемени получила на решение проблемы две субсидии от фирмы «General Electric». Реализация разработок языка и операционной системы осуществлялась с осени 1963 г. на компьютерах фирмы «General Electric» — DATANET — 30, GE-225, GE-235. Первая программа на БЕЙСИКе с использованием системы разделения времени с тремя терминалами заработала в мае 1964 г. (к осени этого года система работала с 12 терминальными устройствами).

Необходимо отметить, что создавался БЕЙСИК не на пустом месте, а на основе существовавших к тому времени алгоритмических языков. Взаимосвязь БЕЙСИКа с ФОРТРАНом и АЛГОЛом очевидна. Из ФОРТРАНа взято указание параметров цикла: начало, конец и шаг. Из АЛГОЛа пришли слова FOR и STEP и более естественная проверка завершения выполнения цикла. Некоторые заимствования пришли из языков IOSS и CORC.

Первая программа на БЕЙСИКе, опубликованная Кемени и Куртцем в 1964 г., имела вид.

```
10  LET X=(7+8)/3
20  PRINT X
30  END
```

БЕЙСИК создавался как язык интерпретирующего типа в отличие от компилирующего ФОРТРАНа.

Возможно, одной из причин создания БЕЙСИКа интерпрети-

быть получен через программные прерывания. Причем часть подпрограмм, входящих в IBMDOS.COM, вызывается через свое собственное прерывание, а часть — через общее прерывание 33. По терминологии DOS первую группу называют прерываниями DOS, а вторую — вызовами функций DOS.

Все услуги, предоставляемые IBM DOSCOM вышележащим слоям, можно разделить на пять категорий: ввод-вывод на стандартные символьные устройства, управление файлами и каталогами, управление памятью, управление процессами и разное.

Работа со стандартными символьными устройствами, к числу которых отнесен дисплей и его клавиатура, принтер и асинхронный адаптер связи, выполняется с помощью функций, весьма похожих на те, что предоставляет нам BIOS.

Различные функции, общим числом двенадцать, отличаются друг от друга устройством, задействованным в операции, размером передаваемых данных: символ или строка, наличием ожидания и эхо-печати символа на экран при вводе.

Управление файлами и каталогами образует самую многочисленную категорию вызовов: их двадцать восемь.

Работа с файлом начинается с создания его или открытия, если он уже существует. Программа передает DOS спецификацию файла и атрибуты, которые должны быть сопоставлены файлу, а DOS возвращает 16-разрядное число, которое служит в дальнейшем для идентификации файла, являясь как бы номером канала, связывающего программу и файл или устройство. Запуск программы сопровождается открытием пяти стандартных каналов: стандартный ввод и вывод, вывод ошибок, вспомогательный ввод-вывод и принтер. Таким образом, номера каналов с 0 до 4 зарезервированы, и для открываемых

рующего типа было то, что большинство мини-ЭВМ тех лет (HP 200. PDP-8, PDP-11, WANG и др.) использовали интерпретаторы.

В период с 1965 по 1971 г. возникли шесть авторских редакций языка БЕЙСИК по мере его усовершенствования. Широкую известность получила редакция БЕЙСИКа, реализованная в 1967 г. на машине GE-400 и несколько позднее (1969 г.) на GE-635.

В дальнейшем (70-е гг.) на основе авторских редакций языка фирмами «Hewlett—Packard», «Digital Equipment», CDC, «Honeywell» и др. были разработаны версии БЕЙСИКа для различных типов мини-ЭВМ.

В начале 70-х годов появились первые микро-ЭВМ. Первые два варианта интерпретатора языка БЕЙСИК для микро-ЭВМ Altair-8800 были созданы сотрудниками фирмы MITS П. Алленом и Б. Гейтсом в 1975 г.

В следующем году П. Аллен и Б. Гейтс основали фирму «Microsoft», которая внесла большой вклад в развитие языка и его реализацию на микро-ЭВМ и персональных ЭВМ. Первоначальные версии языка фирма «Microsoft» в 1976—1978 гг. поставила для ПЭВМ фирм «Commodore», «Apple», «Tandy».

В 1979 г. фирмой «Microsoft» была разработана версия языка — MBASIC (распространенное названий — БЕЙСИК-80), которая получила известность также благодаря созданию компактного интерпретатора и операционной системы MS DOS, предназначенных для персональных компьютеров фирмы IBM, первая модель которых появилась в августе 1981 г. В этом же году для компьютеров IBM PC фирма «Microsoft» представила расширенную версию БЕЙСИКа-80 под названием BASIC-A («Advanced — передовой), которая поддерживала текстовой и графический режимы. В 1984 г. в BASIC-A выведены дополнительные возможности, такие, как работы с окном экрана, обработка прерываний от таймера, выполнение команд операционной системы и пр. В этом же году фирма «Microsoft» разработала интерпретатор языка Macintoch BASIC для ПЭВМ Macintoch фирмы «Apple».

Развитием языка BASIC-A стала версия Quick BASIC, включающая подпрограммы и функции с локальными и глобальными переменными, средства поддержки графики и звука, алфавитно-цифровые метки и т. д. В 1985 г. создатели БЕЙСИКа Т. Куртц и Дж. Кемени разработали для IBM PC версию языка под названием True BASIC.

Существенный вклад фирма «Microsoft» внесла также и в разработку программных средств для бытовых компьютеров. В 1983—1985 гг. токийский филиал фирмы разработал стандарт на 8-разрядные бытовые компьютеры MSX и MSX-2, выпускаемые рядом японских фирм, дисковую операционную систему MSX DOS и интерпретатор языка MSX BASIC. Версия стандарта MSX-2 имеет дополнительные графические возможности.

Что касается стандартизации, то БЕЙСИК последовал «по тяжкому пути» ФОРТРАНа, когда попытки составить стандартное описание были сделаны после долгих 15 лет использования различных версий. Только в 1978 г. был принят стандарт ANSI на минимальный БЕЙСИК (Minimal BASIC—ANSI X3.60), который, однако, не

программой файлов используются ближайшие доступные номера. Правда, имеется возможность переключить любой из пяти каналов на другой файл или устройство.

При работе с файлом он рассматривается как последовательность байт без каких-либо предположений относительно структуры записи или метода доступа. Для чтения или записи данных в файл требуется только указать буфер и количество байт. Обеспечение произвольного доступа производится функцией перемещения указателя положения в файле.

Функции управления каталогом достаточно разнообразны и обеспечивают гибкое управление иерархической структурой системы каталогов, позволяя создавать и удалять подкаталоги, переходить из одного подкаталога в другой, обрабатывать группы файлов и т. д.

Управление памятью реализует небольшой набор функций, достаточный для работы на ПЭВМ, но очень простой по сравнению с многопользовательскими системами. DOS следит за распределением памяти, записывая в начало каждой области памяти специальную таблицу — блок управления памятью. Этот блок содержит размер области, имя владельца, если он есть, и указатель на следующую область.

Когда поступает запрос на дополнительную память, DOS ищет достаточно большую область доступной памяти для удовлетворения запроса. Если она ее находит, то изменяет блок управления памяти, указывая владельца. Если найденная область больше затребованной, то создается новый блок управления памятью в начале ненужной части области с изменением поля размера в блоке управления найденной области и модернизацией цепочки блоков управления памятью. При освобождении области памяти в соответствующий

решил всех проблем спонтанного развития языка. Вот что пишет по этому поводу Р. Форсайт: «Сразу же после своего появления БЕЙСИК стал меняться: с ним начал происходить процесс, похожий на эволюцию естественных языков. Хотя одна из версий была утверждена комитетом стандартов и закреплена разработчиками транслятора, язык вскоре вышел из повиновения и стал общественным достоянием».

Один из создателей БЕЙСИКа Т. Куртц, ставший председателем комитета ANSI по разработке стандарта языка, в 1982 г. опубликовал в журнале BYTE проект расширенного стандарта языка. Проект нового стандарта на язык БЕЙСИК выдвинут совместно с ANSI рабочими группами двух европейских организаций — ECMA (Ассоциация вычислительных средств) и EWICS (Комитет промышленных средств вычислительной техники).

Согласно публикации по новому стандарту (BYTE. — 1988.— 13. — № 9) определены следующие особенности языка: нумерация строк необязательна; введены новые типы данных; разрешены символьные константы; введены новые условные операторы — расширенный IF, оператор выбора SELECT CASE; введен оператор выхода из цикла EXIT FOR и др.

В нашей стране БЕЙСИК в пакетном режиме впервые был реализован в 1969—1970 гг. на ЭВМ типа М-20 сотрудниками Горьковского университета под руководством Ю. Кеткова. Им же была опубликована первая доступная книга по программированию на БЕЙСИКе (см. [2] в списке литературы). Затем появилась модернизация БЕЙСИКа для ЭВМ М-222, которая позволила его использовать в режиме разделения времени. В дальнейшем сотрудники ИК АН УССР и ВЦ Сибирского отделения АН СССР реализовали БЕЙСИК на высокопроизводительной ЭВМ второго поколения БЭСМ-6.

Последующая отечественная история БЕЙСИКа связана с мини-ЭВМ, микро-ЭВМ и ПЭВМ. В мини-ЭВМ М 6000, М 7000, СМ1 и СМ-2 реализована версия БЕЙСИКа, соответствующая первым авторским редакциям конца 60-х годов, а в мини-ЭВМ СМ-3, СМ-4, имеющих в составе операционные системы ОС РВ и РАФОС, используется несколько версий БЕЙСИКа. Версия языка под названием БЕЙСИК-плюс, разработанная в 1975 г. для мини-ЭВМ PDP-11, широко применялась в СМ-3 и СМ-4.

В отечественных ПЭВМ применяются различные диалекты языка БЕЙСИК. Так, в ПЭВМ «Искра-220» используется расширенная версия языка системы WANG-2200 B, а в ПЭВМ «Агат» — интерпретатор языка, идентичный тому, который применяется в ПЭВМ «Apple II». Версия БЕЙСИК-плюс используется в микро-ЭВМ

блок управления памятью вносятся изменения, показывающие, что данная область памяти доступна для распределения.

Прежде чем перейти к рассмотрению функций управления процессами, приведем несколько слов о самой концепции процесса. В самой идее процесса лежит способ рассмотрения и управления программами в ходе их выполнения. Понятие процесса позволяет нам делать различие между статическими командами, расположенными в памяти, и последовательностями действий, которые они порождают при их выполнении. Можно считать доказанной практическую полезность понятия процесса как активного строительного блока для операционных систем и прикладных систем главным образом потому, что он позволяет естественным образом отразить многие логически и физически параллельные действия в системе. Средства управления процессами — это часть операционной системы, которая реализует понятие процессов или задач, выполняющихся параллельно на виртуальных процессорах.

В PC DOS функции управления процессами включают действия по загрузке, выполнению и завершению программ. Прикладные программы также могут воспользоваться этими функциями для управления другими программами.

Когда программа загружает и требует выполнения другой программы, DOS выделяет память, создает в ней специальную таблицу: префикс программного сегмента, начиная со смещения 0 выделенной памяти, загружает программу и передает ей управление. Когда вызванная программа завершается, управление возвращается вызвавшей ее программе.

Как видно, возможности системы в этом вопросе не слишком велики. Разумеется, в этом есть и положительная

БЕЙСИК

«Электроника 60» и диалоговых вычислительных комплексах ДВК-1, ДВК-2, ДВК-3. Для новых учебных вычислительных комплексов типа «Корвет» разработан интерпретатор языка БЕЙСИК на основе версии MBASIC. Профессиональные персональные ЭВМ, такие, как ЕС 1341/42, «Искра 1030.11», «Нейрон И9.66», программно-совместимые с IBM PC, работают с указанными выше версиями БЕЙСИКа в ОС CP/M-86 и MS DOS. Кроме того, в некоторых учебных заведениях страны успешно эксплуатируется несколько тысяч японских бытовых компьютеров «Ямаха MSX» и «Ямаха MSX-2», используемых в качестве комплектов учебной вычислительной техники (КУВТ), в которых программы реализуются на языке MSX-BASIC.

С июля 1989 г. в нашей стране введен стандарт на язык программирования БЕЙСИК (ГОСТ 27787-88).

О популярности БЕЙСИКа среди массового пользования персональных компьютеров говорить не приходится, даже несмотря на утраченную былую его простоту — она велика. И пока спада этой популярности не наблюдается.

В 90-х годах должны появиться компьютеры пятого поколения, о проекте создания которых впервые заявили японские ученые в начале 80-х годов. Основным языком программирования в этом осуществляемом компьютерном проекте предусматривается язык ПРОЛОГ. А как же БЕЙСИК? Любопытные соображения по этому поводу высказал Р. Форсайт: «Если вам необходимо использовать ПРОЛОГ, то подождите, пока он будет «проглочен» языком БЕЙСИК, этим питоном, пожирающим все, что лежит на его пути. БЕЙСИК только что закончил «переваривать» язык ПАСКАЛЬ со всеми его управляющими структурами. После небольшой паузы и нескольких отрыжек он будет в состоянии «слопать» ПРОЛОГ, так что мы увидим варианты языка БЕЙСИК со встроенным механизмом доказательства теорем методом резолюций. Тогда наступит время подумать о возможном переходе на ПРОЛОГ».

Литература

1. Kurtz T. E. BASIC. ACM SIGPLAN NOTICES, 1978. — № 13, 8. — P. 103—118.

2. Кетков Ю. Л. Программирование на БЕЙСИКе. — М.: Финансы и статистика, 1978. — 158 с.

3. Моррил Г. БЕЙСИК для ПК ИБМ / Пер. с англ. — М.: Финансы и статистика, 1987. — 207 с.

4. Корчак А. Е. Язык программирования БЕЙСИК для микро-ЭВМ, — М.: МНИИПУ, 1988. — 131 с.

5. Кетков Ю. Л. Диалог на языке БЕЙСИК для мини- и микро-ЭВМ. — М.: Наука, Гл. ред. физ.-мат. лит., 1988. — 368 с.

6. Блэнд Г. Основы программирования на языке БЕЙСИК в стандарте MSX / Пер. с англ. — М.: Финансы и статистика, 1989. — 208 с.

М. Малыхина,
А. Частиков

черта: проще оказалась разработка системы, меньше накладные расходы (память, процессорное время) при эксплуатации. Но необходимость перезагрузки прикладных систем, предоставляющих различные услуги, необходимые для решения единой задачи, может потребовать несколько лишних часов для того, чтобы эту задачу решить. Действительно, для подготовки отчета вам могут потребоваться текстовый редактор и система управления базами данных, табличный процессор, средства машинной графики и т. д.

Возможны два варианта решения этой проблемы. В первом случае специальная программа, программа управления окнами, разбивает всю рабочую память ЭВМ на сегменты и загружает в каждый такой сегмент одну из нужных вам прикладных программ. Для переключения с одной прикладной программы на другую необходимо лишь указать на нужное окно, соответствующее желаемой программе.

Во втором случае переход также осуществляется через механизм окон. Но в отличие от программ управления окнами программы, обеспечивающие многозадачный режим, организуют работу программ в фоновом режиме. Дело в том, что почти 90% времени процессора тратится на ожидание какого-либо внешнего события, например ввода с клавиатуры. В многозадачном режиме предпринимается попытка использовать это время для выполнения фоновых программ.

Система Top View позволяет организовать такой многозадачный режим, используя метод разделения времени. Все время процессора разделяется на небольшие интервалы, и всем выполняющимся программам поочередно выделяется такой интервал. Интервал времени очень мал, порядка нескольких миллисекунд, и действия по переключению с одной программы на другую проходят незаметно и для прикладной программы, и для пользователя. Так что кажется, что все программы выполняются параллельно, хотя, конечно же, в каждый момент времени выполняется лишь одна программа.

Разумеется, за все приходится платить. И выполнение каждой отдельно взятой программы замедлится по сравнению с тем, когда она выполняется одна. Да и часть памяти придется отдать под Top View.

COMMAND. COM. До сих пор рассмотренные нами возможности в основном доступны только из программы. Для того чтобы пользователь мог непосредственно управлять вычислительной установкой, необходима специальная программа, которая была бы способна считать введенную с клавиатуры команду, определить и выполнить то, что требует от нее пользователь. Такая программа обычно называется командным процессором, и в данном случае им является COMMAND. COM.

Если вы введете какую-либо внутреннюю команду, такую, как DIR, COPY или TYPE, то вы потребуете услугу, которая «встроена» в COMMAND. COM и может быть предоставлена немедленно.

Для распознавания внутренних команд COMMAND. COM содержит таблицу имен команд. Если он не находит очередную команду в этой таблице, то считает ее внешней командой и пытается найти на дискете файл с таким же именем и типом: COM, EXE или BAT. Первые два типа файлов им обрабатываются практически одинаково: загружает их в память и передает им управление с помощью вызовов, обслуживаемых IBMDOS. COM. Файл типа BAT рассматривается командным процессором как текстовый файл, содержащий серию команд, которые и выполняются точно так же, как если бы они были введены с клавиатуры.

Командный процессор COMMAND. COM фактически разбит на три части. Первая часть загружается в память следом за программами из файлов IBMBIO. COM и IBMDOS. COM и подобно им становится резидентной частью DOS.

Вторая часть COMMAND. COM используется только временно, во время начальной загрузки системы, для поиска и выполнения командного файла AUTOEXEC. BAT. После его выполнения эта часть COMMAND. COM стирается.

Третья, наиболее интересная, часть COMMAND. COM содержит интерпретатор команд, включая подпрограммы

для выполнения внутренних команд, и является полурезидентной. С одной стороны, конечно, желательно, чтобы интерпретатор все время находился в памяти, но с другой стороны, в связи с большим количеством выполняемых им функций он занимает достаточно много места в памяти.

Решение этой проблемы весьма любопытно. Эта часть COMMAND. COM помещается в самых старших адресах доступной памяти (обычно эта область памяти используется в последнюю очередь), и разрешается ее перекрывать другими программами. Когда возникает необходимость в использовании интерпретатора команд, резидентная часть COMMAND.COM проверяет, остался ли он все еще неизменным. Если нет, то он повторно загружается с диском.

Основное средство связи с пользователем — дисплей — предоставляет более богатые возможности, чем те, что используются COMMAND.COM. Это было замечено, и разработан ряд программ, которые иначе организуют работу с пользователем, чем последовательное выполнение вводимых с клавиатуры команд. Основу взаимодействия в такого рода программах составляют меню, поддержание нескольких окон.

Например, в программе XTreePro вся информация о дискетах, каталогах и файлах выводится через восемь разделов экрана: окна каталогов, окна файлов, окна команд, окна управляющих клавиш, окна, определяющие текущий диск, спецификацию файлов и статистическую информацию, а также окна функциональных клавиш.

В окне каталогов представляется структура каталогов текущего диска. Корневой каталог изображается наверху списка, а все остальные каталоги отсортированы по алфавиту и изображаются сверху донизу окна. Причем главные каталоги располагаются слева, а подкаталоги — справа.

Если подсвеченный прямоугольник располагается в окне каталогов, то это окно является активным. Когда окно каталогов активно, подсвеченный прямоугольник указывает текущий каталог и доступны каталожные команды.

Окно файлов содержит список файлов, расположенных в текущем каталоге, на данном устройстве или группе устройств в зависимости от выбранного формата окна. В список файлов включаются только файлы, которые соответствуют спецификации, показываемой в окне спецификации файлов. Когда окно файлов активно, подсвеченный прямоугольник указывает текущий файл. Именно на этот файл воздействуют файловые команды XTree Pro.

Переключение между этими двумя окнами производится нажатием клавиши «Ввод».

Выбор команды производится нажатием одной клавиши, выделенной в названиях команд, список которых приведен в окне команд. В каталожные команды входят команды удаления и создания каталогов, переименования каталогов и переключение дисков и другие. В файловые команды входят команды копирования и перемещения, редактирования и удаления отдельных файлов и другие. Вместе с тем имеется возможность отметить группу файлов и произвести действия над всей группой файлов.

Таким образом, XTreePro делает работу с файлами и каталогами нагляднее, упрощает и ускоряет эту работу, сокращая объем ввода.

Что дальше?

Хотя рассмотренная нами операционная система и не включает всех механизмов и методов, применяемых в современных операционных системах, на ее примере видны тенденции развития операционных систем.

Эволюция структуры и организации операционных систем привела к замене ранних монолитных систем на системы с четко выделенным ядром. В таких системах ядро реализует основные операции над ресурсами, располагаясь непосредственно над аппаратурой, трансформирует машину в виртуальную ЭВМ, в которой процессы более высокого уровня, другие объекты реализуются существенно проще. Относительно неструктурированные конструкции уступили дорогу иерархически орга-

низованным системам, состоящим из нескольких последовательных слоев, каждый из которых привносит для пользователей новый уровень услуг, более совершенных, чем ранее. Многие ранние системы были замкнутыми в том смысле, что множество предоставляемых ими услуг фиксированы и было очень непросто добавить туда новые. Многие новые системы являются открытыми, обеспечивая пользователям возможности по их расширению.

Достаточно очевидно, что все упомянутые черты присущи PC DOS: и выделение ряда уровней, и возможность расширения как за счет новых драйверов устройств, так и за счет внешних команд. Разумеется, далеко не все вопросы, связанные с операционными системами, нашли здесь свое отражение в полной мере, например вопросы организации распределенных систем, параллельной обработки данных. Но хочется надеяться, что изложенный здесь материал поможет читателям в их практической работе, упорядочит представления о месте и возможностях операционных систем, упростит знакомство с новыми системами. Развитие микропроцессорной техники, появление 32-разрядных процессоров сказались и на развитии ОС. Новая система OS/2, ориентированная на такого рода процессор, уже начала распространяться среди пользователей персональных ЭВМ. Естественно, что она обеспечивает ряд новых возможностей, которые и призваны привлечь к ней внимание:

— обеспечивает поддержку адресного пространства до 16 Мбайт;

— осуществляет параллельное выполнение нескольких прикладных систем, позволяя организовывать параллельные процессы внутри прикладных систем;

— представляет более совершенный интерфейс как прикладной программный интерфейс за счет его расширения и увеличения гибкости посредством механизма динамических библиотек, так и интерфейс с пользователем, обеспечиваемый менеджером представлений.

Переход DOS—OS/2 может быть выполнен постепенно, многие прикладные системы перенесены в OS/2 без изменения интерфейса. Разработаны программные средства, обеспечивающие работу в смешанной сетевой среде, где присутствует и DOS и OS/2, например сервер базы данных, поддерживающий язык SQL, совместная разработка фирм «MicroSoft» и «Ashton-Tate». Кроме того, старшие версии DOS имеют многие черты, сходные с OS/2.

В августе 1981 г. фирма IBM объявила о выпуске персонального компьютера PC AT, чем ввела своих конкурентов в состояние шока.
Основной операционной системой PC AT явилась система MC DOC, разработанная фирмой «Microsoft». Эта операционная система поставляется на все персональные компьютеры, совместимые с PC AT по лицензии фирмы «Microsoft».

В. Я. ЦВЕТКОВ

Операционная система MC — DOC

Носителем операционной системы MC DOS для большинства моделей персонального компьютера PC служат гибкие магнитные диски или дискеты. Большая часть дискеты предназначена для хранения данных или программ. В небольшой части хранятся оглавление (каталог) программ пользователя и операционной системы, а также информация о самой дискете (номер тома, тип версии).

Загрузка операционной системы MC DOC происходит после включения ЭВМ. На первом этапе срабатывает программа, встроенная в персональный компьютер. Ее назначение — самопроверка оперативной памяти и чтение начала дискеты, на которой находится загрузочная программа операционной системы. На втором этапе считанная программа — загрузчик, в свою оче-

редь, запускает операционную систему МС DOC.

На первом этапе она выводит текущую дату на терминал, например: Current date is Tue 10-10-1988, и запрашивает информацию на установку новой даты.

Новая дата вводится через разделительную черточку «-» в последовательности «день-месяц-год».

После этого выдается сообщение о текущем времени, например Current time is 0:00:12.34, и запрашивается информация для установки правильного времени Enter new time.

После этих процедур на дисплей выводится сообщение о типе загружаемой операционной системы и номер ее версии и указывается активизированный диск «А» или «В».

Операционная система МС DOC обеспечивает работу в интерактивном или пакетном режиме. Большой интерес представляет интерактивный режим, который рассматривается ниже.

Одной из важных и часто употребляемых процедур является активизация диска. В стандартном исполнении персональный компьютер имеет два диска. Один из них всегда активный, т. е. по умолчанию (без названия диска в имени файла) чтение и запись файлов осуществляются с этого диска. Эти диски обозначают буквами «А» и «В». После загрузки, как правило, активным является диск «А». Для переназначения оператор набирает на пульте дисплея команду

B:

или

B>

в зависимости от версии операционной системы.

Для работы с новыми дискетами их необходимо сформатировать. Если мы хотим форматировать новую дискету, установленную, например, на дисководе «В», для этого набирается команда

Format B: — S.

Первая часть команды вызывает программу форматирования, вторая часть указывает тип дисковода с форматируемой дискетой, третья часть (необязательная) указывает на необходимость получения копии МС DOC на вновь сформатированной дискете.

После завершения форматирования на дисплей выводится запрос о продолжении форматирования с подсказкой:

Format another ?

В случае продолжения форматирования набирается символ «Y», при завершении — символ «N».

Дальнейшая часть работы заключается в создании программы на алгоритмическом языке, записи ее на диск как исходного модуля, получении объектного модуля, получении загрузочного модуля, запуска и отладки программы. В том случае, если используется интерпретатор, например БЕЙСИК, все названные процедуры совмещаются.

Прежде чем написать программу, необходимо выбрать для нее имя файла. Имя файла в МС DOC, как и во многих операционных системах, состоит из двух частей. Первая часть — собственно имя файла — может состоять из восьми или меньшего числа символов. Вторая часть — расширение имени файла — отделяется от имени файла точкой и может содержать не более трех символов. Некоторые виды расширений носят служебный характер, что должно учитываться пользователем при составлении программ и при работе с файлами.

В табл. 1 приведены основные служебные расширения имен файлов.

Таблица 1

Основные расширения файлов в операционной системе МС DOC

Расширение	Функция
ASM	Расширение программ, написанных на АССАМБЛЕРе
BAK	Расширение копии программы после редактирования сигнала
BAS	Расширение программ, написанных на языке BASIC
BAT	Пакетно обрабатываемые данные
C	Расширение программ языка Си
COD	Расширение листинга объектных модулей после компиляции
COM	Расширение выполняемой программы
DAT	Файлы данных
EXE	Выполнение программы в перемещаемом формате

Расширение	Функция
FOR	Расширение программ, написанных на ФОРТРАНе
LIB	Расширение файлов библиотек
LST	Расширение листингов
MAP	Специальные файлы, получаемые после линкования
OBJ	Расширение объективного модуля
PAS	Расширение программ, написанных на ПАСКАЛе
TMP	Расширение временных файлов

Для написания текстов программ на любом языке программирования используют программу — редактор текста. Эта программа в МС DOC носит название EDLIN.

Вызов его осуществляется с функциональной клавиатуры дисплея с указанием программ редактора и редактируемой программы, например:

EDLIN B:FYLE. FOR
B:EDLIN A:FYLE. FOR

В первом случае редактор вызывается с дискеты «А», а программа — с дискеты «В». Во втором — наоборот.

После названия редактора указывается носитель и имя файла. Если файл с указанным именем не найден на дискете, выдается сообщение «новый файл». Если файл найден, он вводится в оперативную память и после завершения ввода выдается сообщение

«Конец ввода файла»

Если файл настолько велик, что в оперативной памяти компьютера места для него недостаточно, то сообщение о конце ввода не выводится. Файл в этом случае вводится частично, насколько позволяет оперативная память. Такие файлы редактируются по частям, отредактированная часть записывается на диск, а на свободное место заносится неотредактированная часть файла. Команды редактора текста приведены в табл. 2.

Таблица 2

Основные команды редактора текста

Команда	Функция
A	Повторное чтение с диска оставшейся части файла
C	Копирование одной части в другую
D	Стирание строк
E	Завершение редактирования и запись в файл вывода выход из редактора
I	Запись (дополнение) данных
L	Выдача строк с номерами на дисплей
M	Перемещение части данных из одного места программы в другое
P	Страница. Служебная команда для удобства выдачи
Q	Завершение редактирования без записи на диск
R	Поиск образца текста с заменой на новый образец
S	Поиск образца текста и локализация курсора в этом месте
T	Передача всего содержимого разных файлов без прекращения и завершения редактирования
W	Запись части редактированного текста в файл, вывод на диск без завершения редактирования

После вызова редактора можно вводить или корректировать текст. Редактирование в МС DOC осуществляется построчно.

Основной командой для ввода текста строки служит команда «I». Ее формат NI.

N — номер строки, на которой вводится текст.

Противоположной по значению командой является команда D. Она имеет форматы: ND; N1, N2D; D; N2D; N1, D.

В первом случае стирается строка с номером N. Во втором стираются строки с номерами от N1 до N2 включительно. В третьем случае стирается текущая строка. В четвертом случае стираются строки от текущей до указанной, в последнем — от указанной до текущей.

По команде L на экран выводится несколько строк. Возможны два варианта:

NL

или

N1, N2L.

В первом случае будет выведено N строк, начиная с текущей. Во втором — будут выведены строки с номерами от N1 до N2 включительно.

Команда S позволяет осуществлять поиски по образцу, формат команды

N1, N2 SNAME

NAME образец, который ищется в тексте, начиная от строки с номером N1 до строки с номером N2.

Разновидностью команды S является команда R. По этой команде старый образец ищется и заменяется на новый. Формат команды

N1, N2 R OLDNAME Z NEWNAME

N1, N2 — номера строк, в области которых ищется образец, стоящий после команды R. После Z стоит новый образец, который заменяет старый.

При необходимости записать N строк текста на диск применяется команда NW.

Для добавления строк в рабочую область существует команда A, которая имеет форму NA.

N — количество строк, которые считываются с диска. Если считывается конец файла, выдается сообщение о конце входного файла.

В версии 2 редактор текста имеет дополнительные команды, наиболее важные — это C, M, T. Команда C позволяет производить многократное копирование части текста в заданном месте. Команда M осуществляет ту же операцию однократно. Команда T вводит данные другого файла в рабочую область.

Хотя название программы может быть произвольным и его определяет пользователь, это имя не должно совпадать со служебными названиями операционной системы. Последние включают команды операционной системы (табл. 3) и служебные названия устройств. К последним относятся:

NUL — нуль устройства

CON — консоль пользователя

USER — пользователь, что служит альтернативным обозначением консоли

AUX — канал связи, например, асинхронный

COM I — первый из возможных каналов связи (аналог АИХ). COM2 и COM3 имеют специальное назначение

PRN — устройство вывода

PRT — устройство вывода

Программа, написанная на алгоритмическом языке, преобразуется с помощью интерпретатора или компилятора. Обычно для алгоритмического языка поставляется либо транслятор, либо компилятор. Особенностью операционной системы МС DOC является то, что алгоритмический язык БЕЙСИК поставляется и с интерпретатором, и с компилятором.

Кроме этого языка, в МС DOC существует возможность написания программ на алгоритмических языках: макроАССЕМБЛЕР, ФОРТРАН, КОБОЛ, ПАСКАЛЬ, Си, ФОРТ, ЛИСП, ЛОГО, АПЛ, АДА, СМОЛЛТОК и др.

Следует отметить, что системные средства МС DOC позволяют переводить программы, написанные на ФОРТРАНе, в программы на языке БЕЙСИК. Для программ с высокой степенью структурированности данных рекомендуются языки ПАСКАЛЬ и Си. Следует напомнить, что ПАСКАЛЬ позволяет включать фрагменты текстов, написанные на макроАССЕМБЛЕРе. Это делает его предпочтительным при работе с регистрирующими внешними устройствами. ПАСКАЛЬ и Си рекомендуются для работы программистам, имеющим достаточно высокую квалификацию. Для остальных целесообразно применение ФОРТРАНа и особенно БЕЙСИКа.

Компилятор переводит текст программы (исходный модуль) в объектный файл, который получает расширение OBJ. Интерпретатор позволяет одновременно с преобразованием осуществлять процесс вычислений.

Для получения загрузочного модуля из нескольких объектных модулей, а также с использованием библиотек применяется программа LINK. Входными для нее являются файлы с расширениями OBL и LIB, а выходными — загрузочные файлы в так называемом перемещаемом формате. Загрузочные файлы имеют расширения EXE и COM. Причем после линкования программа всегда получает расширение EXE. Она может быть конвертирована в программу с расширением COM с помощью команды EXE2BIN (см. табл. 3). Основное различие между этими командами в том, что программа с расширением COM может загружаться в паузах программы с расширением EXE. Иными словами, EXE соответствует программам оперативного режима, COM — фонового.

Запуск программ осуществляется вызовом файла с расширением EXE или COM.

В табл. 3 представлены только основные команды версии 1. Версии MC DOC 2 и 3 имеют ряд дополнительных команд. Все команды версии 1 входят как основа в версии 2 и 3.

Таблица 3

Список основных команд MC DOC

№	Команда	Функция
1	COMP	Сравнение двух файлов с указанием различий
2	COP	Копирование файлов
3	CREF	Формирование таблицы перекрестных ссылок
4	DATE	Изменение текущей даты
5	DEBU6	Вызов программы-отладчика
6	DEL	Стирание файлов с дискеты
7	DIR	Распечатка каталогов дискеты
8	DISKCOMP	Сравнение двух дискет
9	DISKCOPY	Копирование дисков
10	EDLIN	Вызов программы редактор текстов
11	EPASE	Стирание файлов
12	EXE2BIN	Конвертирование программ из формата EXE в формат COM
13	EXEFIX	Конвертирование программ из формата COM в формат EXE
14	FC	Сравнение файлов
15	FILECOMP	
16	FORMAT	Формирование дискет
17	LIB	Работа с библиотеками
18	LIMK	Создание загрузочного модуля
19	LOCATE	Аналог EXE BI
20	MASM	Вызов транслятора макроассемблера
21	PRINT	Вывод файлов на устройство печати
22	PAUSE	Ожидание ответа с клавиатуры (пакетный режим)
23	EM	Переименование файлов
24	RFM	Переименование файлов
25	SYS	Копирование скрытных служебных файлов
26	TIME	Установка времени
27	TYPE	Вывод на дисплей файлов с расширением TXT
28	VER	Выдача номера версии
29	VERYVY	Проверка правильности записи на диск
30	VOL	Выдача информации об оглавлении тома (дискеты)

Для копирования отдельных файлов или групп файлов используется программа COPY. Например, для копирования файла FYLEI. MAC с дискеты «А» на дискету «В», где ему будет присвоено новое имя FNLEZ. MAC, необходимо набрать команду COP A:FYLE.MAC B:FYLE2.MAC.

Двоеточие отделяет носитель от имени файла.

Для копирования всех файлов с «А» на «В» используется команда COPY A:**B:

Звездочка соответствует смысловому значению «любой». Программа COPY во многом аналогична программе COPY в операционных системах РАФОС, ОС РВ.

Для вывода или изменения текущих даты и времени применяют команды DATE и TIME. Для вывода после набора команды нажимают возврат каретки, для изменения после набора команды через пробел вводят новую дату или время. Для даты в качестве разделителя используют черточку «-», для времени — двоеточие «:».

Для копирования только служебных «скрытых» файлов операционной системы MC DOC используют команду SYS. Эта программа копирует только системные программы.

При ошибке в наборе команды или при наборе команды, отсутствующей в используемой версии, MC DOC выдает сообщение «неправильная команда или файл».

Для проверки содержимого каталога дискеты или наличия отдельного файла на ней применяют команду DIR.

Например:

DIR A: TEST.FOR

или

DIR B:

В первом случае на терминал будет выведено сообщение о файле TEST.FOR, находящемся на дискете «А», с указанием его размеров и даты создания. Во втором случае будет выдан каталог дискеты «В».

Объектные файлы могут объединяться в общие файлы — библиотеки, каждая из которых имеет собственный каталог.

Библиотеки в MC DOC могут иметь «древообразную» структуру (рис. 1), т. е. допускать вложение одних библиотек в другие.

Древовидные структуры файлов могут создаваться как в рамках одной дискеты, так и в нескольких.

При необходимости вывода текста программы на устройство печати используют команду PRINT. Например,

PPINT A:TASK.BAS

выводит текст программы TASK, написанной на алгоритмическом языке БЕЙСИК, с диска «А».

Для переименования файлов применяют команду REN. Например, команда

REN FYLEOLD ASM FYLENEW. ASM

переименовывает файл FYLEOLD в FYLENEW с тем же расширением.

Для отладки программ МС DOC содержит отладчик DEBUG. Он позволяет запускать программу с разных точек программы, проверять, менять содержимое ячеек, получать распечатки и т. п., т. е. его работа и возможности аналогичны отладчикам таких операционных систем, как РАФОС, ОС РВ, РОС РВ.

Таблица 4

Основные команды МС DOC

№	Команда	Функция
1	C	Сравнение двух полей в рабочей области и сообщение о различии
2	D	Вывод содержимого поля в шестнадцатиричном и знакоцифровом форматах
3	E	Замена данных в рабочей области
4	F	Вывод данных в рабочей области
5	G	Запуск программы с указанного адреса
6	M	Сложение или вычитание шестнадцатиричного числа
7	I	Чтение каналов ввода-вывода
8	L	Чтение файла диска в рабочую область
9	M	Перемещение данных из одной области в другую
10	N	Объявление имени файла (для команды)
11	O	Передача данных канал ввода/вывода
12	Q	Завершение работы с отладчиком и выход в операционную систему
13	P	Показывает и изменяет содержимое регистров
14	S	Поиск в рабочей области заданных значений адресов
15	T	Трассировка
16	U	Ассемблирование, выдача содержимого рабочей области в форме команд ассемблера
17	W	Запись данных из рабочей области на диск

Проблемы адаптации программного обеспечения

Совершенствование вычислительной техники, например, при переходах от I очереди ЕС и СМ ЭВМ ко II характеризуется не только увеличением оперативной памяти и возрастанием быстродействия, но и совершенствованием систем программирования. В частности, более совершенные типы ЭВМ комплектуются более совершенными типами операционных систем. Это обусловливает переработку созданного ранее программного обеспечения применительно к новому типу операционной системы.

На ЕС ЭВМ такими операционными системами являются DOC, ОС, СВМ. По входным языкам DOC и ОС совместимы частично. Это означает, что исходные модули, составленные для ДОС, могут быть перетранслированы в ОС.

По объектным модулям DOC и ОС несовместимы. Это означает, что объективные модули, подготовленные системой DOC, не могут выполняться под управлением ОС, и наоборот. Несовместимость объектных модулей связана с различием в составе и функциями управляющих программ ОС и DOC.

По-иному дело обстоит при переходе к СВМ. По сравнению с названными СВМ ЕС обладает улучшенными эксплуатационными характеристиками, повышенной надежностью, дополнительными функциональными возможностями, удобными интерактивными режимами. СВМ создает условия для сосуществования объективных программ, написанных в ДОС и ОС, в едином вычислительном процессе.

Аналогичная ситуация имеет место в среде СМ ЭВМ. Программы, составленные в операционных системах ДОС, ФОБОС, РАФОС или ДОС, ОС РВ, совместимы снизу вверх, но не наоборот. В то же время расширенные функциональные возможности более совершенных операционных систем делают необходимым переработку программного обеспечения. Этого требует несовпадение номеров и наименований устройств ввода-вывода для разных операционных систем.

Несовместимость программ для РАФОС и ОС РВ частично устраняется

применением ДОС КП или РОС РВ, операционных систем, допускающих сосуществование программ, составленных в РАФОС и ОС РВ. Но в целом это не решает задачи переработки программного обеспечения.

Таким образом, переход к новой операционной системе или к другому типу ЭВМ влечет необходимость непроизводительной переработки уже созданного и отлаженного программного обеспечения. Трудоемкость этой переработки тем больше, чем менее адаптивно программное обеспечение.

В условиях постоянного обновления ЭВМ (каждые 4—5 лет) и операционных систем с учетом увеличения стоимости программного обеспечения проблема адаптивности и переноса программного обеспечения получила название «проблемы мобильности». Вся история развития программного обеспечения связана с этой проблемой.

Операционные системы, разрабатываемые для решения проблемы мобильности, получили название мобильных. Машинная независимость является главным свойством мобильных систем. Такая независимость предполагает разные типы программно несовместимых ЭВМ.

Мобильные системы снимают проблему переработки программного обеспечения. В нашей стране примером такой системы служит ИНМОС — инструментальная мобильная операционная система. Наибольшую популярность завоевала мобильная операционная система UNIX.

Операционная система UNIX

Операционная система UNIX разработана в известной фирме «Bell Telephone». В той самой «Bell Telephone», где спустя два года после создания первого компьютера будущие лауреаты Нобелевской премии 1956 г. Дж. Бардин, У. Браттейн, У. Б. Шокли создали первый транзистор. В недрах «Bell Telephone» с 1929 г. проводились исследования сигналов из космоса, благодаря чему в 1965 г. два будущих лауреата Нобелевской премии 1978 г. А. Пензиас и Р. Вильсон открыли реликтовое излучение. В этой же «Bell Telephone», где будущие лауреаты Нобелевской премии Л. Шавлов и Ч. Таунс изобретали лазер, была создана операционная система UNIX.

Разрабатываемая с 1969 г. Кеном Томасом и Денисом Ритчи UNIX первоначально предназначалась для проведения исследовательских работ и обработки текстовых документов для внутренних нужд «Bell Telephone».

Головная фирма AT&T, куда входит «Bell Telephone», не является ни производителем ЭВМ, ни ведущим разработчиком программного обеспечения. Популярность и широкое распространение ОС UNIX получила благодаря ее возможностям и преимуществам по отношению к другим ОС.

Широкое использование UNIX дало и отрицательные результаты. В настоящее время имеется свыше 20 поддерживаемых версий этой системы. Из них можно выделить три основных направления: версия, называемая Система V, поддерживаемая корпорацией AT&T; XENIX, основанный на ранней версии; система III, поддерживаемая корпорацией «Microsoft»; версия IV, поддерживаемая Калифорнийским университетом, и др. Эти три направления мало совместимы друг с другом, что вредит репутации системы и создает препятствие для ее развития.

В настоящее время есть основания считать решение этой проблемы делом ближайшего будущего.

Операционная система UNIX устанавливается на ЭВМ с использованием системных средств. В распоряжение пользователя в отличие от МС DOC она поступает готовой: «под ключ». Этим пользователь избавляется от загрузки, активизации носителя системы и т. п. При использовании рабочего терминала для входа в ОС набираются две клавиши:

Control + d

В качестве подтверждения готовности к работе ОС выводит на терминал сообщение:

login:

Это сообщение требует, чтобы пользователь ввел в систему свое условное имя (код), под которым он зарегистрирован.

Имена (коды) всех пользователей, которым разрешена работа, предварительно заносятся администратором (менеджером) системы. Это мероприятие необходимо, чтобы ограничить вход в систему посторонним лицам, поскольку UNIX — многопользовательская система и вход в нее возможен с рабочих терминалов, удаленных терминальных станций и по линиям телефонной сети.

При наборе неправильного имени система ответит знаком вопроса «?» и повторит сообщение. При правильно набранном имени система выводит символ $.

Кроме имени, пользователь может иметь свой пароль, который он устанавливает после входа в систему. Для создания или применения пароля набирается команда

pass

После этого в ответ на запрос UNIX пароль вводится дважды, причем вывод его на дисплей не происходит. Следует отметить, что все команды пользователь завершает символом «ВК» — возврат каретки.

Если пароль у пользователя уже имеется, то на этапе входа в систему он должен сообщить свое имя и пароль.

После этого пользователь может работать с монитором, создавать программы, транслировать, запускать их на счет.

Прежде чем создавать программу, пользователь должен выбрать имя файла для создаваемой программы. Имя файла в UNIX может содержать до 14 символов. Следует отметить, что система различает заглавные и строчные буквы. Пользователь должен помнить, что набор команд и имен файлов осуществляется с использованием строчных (маленьких) букв латинского алфавита.

Имя файла может состоять из двух частей, как в МС DOC. Большая часть, собственно имя, может содержать до 10 символов, меньшая отделяется точкой и может содержать до трех символов. Меньшая часть в руководствах по системе UNIX называется суффиксом, хотя по смысловому значению она является аналогом расширения файла в МС DOC. В качестве примеров суффиксов файлов можно привести:

C — суффикс файла исходного модуля программы, написанной на языке СИ
bac — суффикс программ, написанных на БЕЙСИКе
for — суффикс программ, написанных на ФОРТРАНе
o — суффикс объектного файла
out — суффикс исполняемого файла
sh — суффикс программ интерпретатора

Основные этапы работы с системой заключаются в написании программ, компиляции, запуске, создании файловой структуры, работы с интерпретатором.

Для создания новых текстов программ и изменения старых в операционной системе UNIX имеется редактор текста ed. Как и в МС DOC, он ориентирован на построчное редактирование текстов.

Редактор работает с любым терминалом, генерирующим код ASCII.

Вызов редактора возможен в двух случаях: при создании нового файла и при редактировании старого. В обоих случаях набирается имя редактора, через один или несколько пробелов — имя файла и нажимается клавиша возврата каретки. Например:

ed file в к

Если создается новый файл, то редактор выдает сообщение:

? file

Это означает, что данный файл пустой.

Если открывается существующий файл, то подтверждением этого будет выдача цифрового количества символов, содержащихся в этом файле. Например,

вызов: ed oldfile
сообщение: 1464

Для работы в режиме редактирования пользователю предоставляется ряд команд, которые приведены в табл. 5.

Дополнительно к командам используются метасимволы (табл. 6), расширяющие возможности редактирования.

Использование некоторых команд требует пояснения. Например, команда

Таблица 5

Основные команды редактора операционной системы UNIX

№	Команда	Функция
1	A	Добавление нового текста после текущей строки
2	C	Замена строк текста
3	d	Удаление строк текста
4	/ /q	Многократная замена текста
5	i	Вставка текста перед текущей строкой
6	l	Полная печать со служебными символами
7	lp	Выход на начало файла и печать первой строки
8	m	Перемещение текста
9	P	Печать строки
10	1,$p	Печать всего файла
11	.=П	Печать номера текущей строки
12	q	Выход из редактора
13	S	Замена части текста на новый
14	U	Восстановление строки
15	W	Запись текста в файл
16	$p	Выход на конец файла и печать последней строки
17	/. . . . . ./	Поиск по образцу текста
18	/ /	Повторение поиска
19	BK	Повторение предыдущей команды с переходом к следующей строке команды
20	.BK	Выход из текущей команды
21	A	Добавление текста в начало строки
22	$	Добавление текста в конце строки

Таблица 6

Метасимвол редактора

№	Метасимволы	Значение
1	• (точка)	Один произвольный символ
2	* (звездочка)	Любая последовательность символов
3	[ ] (кв. скобки)	Несколько символов
4	& (амперсанд)	Сокращение набора текста в команде
5	$	Конец файла или строки (по контексту)
6	Λ	Начало строки

печати P в зависимости от применения может иметь разные форматы.

.P — печать текущей строки
I,$P — печать файла с первой строки до последней
N1,N2p — печать, начиная со строки с номером N 1 по строку с номером N 2
.-P — печать только номера текущей строки
.,$P — печать файла с текущей строки до последней
.+NP — пропуск N строк после текущей (к концу файла) и печать строки
.−NP — пропуск N строк перед текущей (к началу файла) и печать строки
$−6,$p — печать 6 последних строк файла
.,$−10p — печать от текущей строки до десятой до конца строки файла

Требует пояснения команда замены S. Формат команды:

S/old text/Newtext/

После самой команды образец текста, подлежащий замене, выделяется символами "/", новый фрагмент текста также завершается символом "/".

Перед использованием команды необходимо найти нужную строку с заданным образцом текста с помощью команды

/text/

Если необходимо найти еще одну сторону с этим же фрагментом, то набирается команда

// в к

Каждый раз после набора команды будет распечатана вся строка, содержащая искомый фрагмент текста.

Различные варианты применения допускает команда удаления d:

1,$d — удаление всего файла с первой по последнюю строку
.,$d — удаление с текущей по последнюю строку
N1,N2d — удаление строк с номерами от 1 по 2 включительно
.+Nd — удаление N-й строки после текущей
.−Nd — удаление N-й строки перед текущей
$−6,$d — удаление 6 последних строк файла

Команда m — перемещения позволяет вставлять фрагмент текста в любое место. Формат этой команды:

N1, N2m N3

N1, N2 указывают номера начала и

конца строк перемещаемого текста. N3 указывает номер строки, после которой будет помещен текст.

Проводя сопоставительный анализ возможностей команд редакторов МС DOC (табл. 3) и UNIX (табл. 5, 6), видно, что возможности редактора UNIX много шире, а сложность работы с обеими практически равная. Это определяет преимущество системы UNIX. UNIX предоставляет в распоряжение пользователя большой комплекс системных программ — утилит. В табл. 7 приведены наиболее часто употребляемые утилиты.

Из сравнения табл. 7 с аналогичной

Таблица 7

№	Утилита	Функция
1	at	Выполнение командного файла в установленное время
2	cat	Объединение (конкатенция) файлов
3	cd	Назначение текущего каталога
4	chmod	Изменение кода защиты файла
5	chown chgrp	Изменение идентификатора владельца или группы
6	CP	Копирование файлов
7	crypt	Шифрование файлов
8	date	Установка времени и даты
9	diff	Сравнение файлов
10	echo	Вывод аргументов
11	file	Определение типа файла
12	find	Поиски файлов
13	grep	Поиск по шаблону
14	kill	Пересылка сигналов
15	ln	Создание связи с файлом
16	lpr	Системная печать
17	ls	Печать содержимого каталога
18	man	Вывод оперативной информации из руководства по операционной системе
19	mail	Организация «почтовой» службы
20	mkdir	Создание каталога
21	mv	Переименование файла
22	nice	Выполнение процесса с указанным приоритетом
23	no hup	Выполнение команды с защитой от прерывания
24	od	Вывод файла в заданном представлении
25	passwd	Изменение пароля
26	pr	Вывод файла на печать
27	ps	Вывод информации о процессах на печать
28	pwd	Печать имени текущего каталога
29	rm	Удаление файла
30	rmdir	Удаление подкаталога
31	sort	Сортировка файлов
32	spell	Поиск орфографических ошибок
33	stty	Установка режима терминала

продолжение табл. 7

№	Утилита	Функция
34	tail	Вывод остатка файла
35	tee	Дублирование стандартного ввода
36	time	Определение времени выполнения процесса
37	tty	Печать имени специального файла терминала
38	wc	Подсчет числа строк, слов или символов
39	who	Выдача информации о пользователях
40	write	Установка двухсторонней связи с другим пользователем

табл. 3 для МС DOC видно, что и здесь система UNIX представляет пользователю значительно более широкие возможности.

Поскольку операционной системе UNIX уделяется достаточно много внимания, в отечественной литературе нет смысла подробно разъяснять каждую команду из приведенных в табл. 7. Процесс компиляции программ в UNIX незначительно отличается от компиляции в МС DOC. Результатом компиляции может быть исполняемый объектный файл a.out. Он аналогичен файлу main, создаваемому при компиляции фортранных программ на ЕС ЭВМ.

Для компиляции используется команда «СС». Пример:

CC tack.C

В результате команды компилируется исходный файл tack, написанный на языке Си, и результатом является файл a.out.

Для назначения имени файлу, полученному в результате компиляции, используется параметр «—0». Пример 2:

CC —0 task task.C

Результат компиляции будет записан в файл task

Для частичной компиляции (только одного файла) используется параметр «—С». Пример 3:

CC —C task task.C

Результат компиляции будет помещен в неисполняемый файл task.o в

отличие от исполняемого файла task в предыдущем примере.

Для создания исполняемого файла из объектных неисполняемых используется команда ld, аналог команды link в MC DOC. В UNIX многие компиляторы, например Си (см. пример 2), вызывают ld.

Возможна компиляция нескольких файлов разного вида. Пример 4:

CC file A.c file B.o file C.c

В приведенном примере два файла file A.c, file C.c, написанные на языке Си, и один объектный файл file B.o компилируются в исполняемый файл a.out.

Ниже приводится практический пример создания программы на языке Си, ее компиляции и запуска. Пример 5:

5.1. С помощью редактора создать файл с именем task.c

5.2. Занести текст программы на языке Си

```
(x перевод из градусов в радианы для
10 значений углов x)
main (      )
{
int low
int upper, st
float grod, rad, ro;
low =1; (x нижний предел градусов x)
upper=10; (x верхний предел градусов
x)
st=1; (x шаг изменения градусов x)
ro=3.1415926/180. 0;
grod=low;
while (grad<=upper){
rad=grad*ro;
printf (“%4.1f% 6.4f\n,grad,rad);
grad=grad+st;}
}
```

Пояснение к программе. Программа переводит 10 значений градусов в радианы и печатает значения градусов и соответствующих им радиан. Комментарии заключаются в (x ... x).

Первый оператор определяет программу как головную (исполняемую). Скобка в строке открывает текст программы. Оператор int определяет целые переменные low, upper, st.

Оператор float определяет переменные с плавающей запятой, grad, rad, ro, while определяет цикл работы программы, пока значение grad не будет больше верхнего предела upper. Оператор printf обеспечивает печать переменных grad, rad в форматах с одной и четырьмя цифрами после десятичной точки. Параметр «\n» служит для перехода на новую строку после печати пары значений grad, rad.

5.3. Записать программу в выходной файл.

5.4. Оттранслировать полученный файл с помощью команды

cc task.o

Если ошибок нет, сообщений об ошибках не появилось — создан файл a.out

5.5. Запустить программу на счет командой

a.out

5.6. В результате работы программы на устройство вывода будут выведены следующие данные:

1.0	0.0174
2.0	0.0349
3.0	0.0524
4.0	0.0698
5.0	0.0873
6.0	0.1396
7.0	0.1222
8.0	0.1396
9.0	0.1571
10.0	0.1745

Одним из специфических и важных понятий системы UNIX является понятие «процесса». Под процессом понимают программу на этапе ее выполнения. Несмотря на то что на самом деле команды программы выполняются процессором ЭВМ, концепция UNIX переносит это действие на процесс. Процесс может быть порожден, активизирован, уничтожен, прерван.

Все работы в системе представляются как множество независимых или взаимосвязанных процессов.

Одним из основных элементов UNIX является ядро системы. Взаимодействие ядра системы и процессов определяет основу функционирования операционной системы. После запуска системы инициализируется ядро, инициализируется система и переходит последовательное иерархическое

порождение процессов. Порождение процессов осуществляется в процессе работы пользователя.

Совокупность взаимодействующих процессов обеспечивает многопользовательский режим работы системы UNIX. Следует подчеркнуть, что эта система не предназначена для работы в реальном масштабе времени, как, например, РАФОС.

Взаимосвязанным с понятием процесса является файловая система UNIX. Она имеет многоуровневую иерархическую структуру. Кроме имени файла, которое создает пользователь, каждый файл имеет особый числовой код — индексный дескриптор, который создает сама операционная система. Индексный дескриптор определяет размер файла, его местонахождение, идентификатор владельца, дату создания, код защиты, метод доступа к файлу, информацию о родительском каталоге файла и т. п.

Система работает с файлом только по индексному дескриптору. Сам дескриптор обычно скрыт от пользователя, и пользователь работает с каталогами файлов, файл может иметь свой каталог.

Типичные периферийные устройства, которые использует операционная система UNIX: диски, ленты, каналы, связи, печатающие устройства. Ввод-вывод информации может происходить байтами или блоками. Блок включает 512 байт. Блочный обмен имеет более высокую скорость ввода-вывода за счет уменьшения обращения к устройству (диски, ленты).

Байтовый ввод-вывод используется при работе с устройствами, не обладающими блочной структурой (терминалы, каналы связи, устройства печати).

Настройка на любую реальную конфигурацию периферийных устройств осуществляется в UNIX программным путем. Она заключается в модификации уже существующих в системе специальных программ. При использовании языка СИ для этой цели служит программа conf.c.

Следует отметить, что система UNIX совершенствуется и приобретает новые достоинства. Первоначально она создавалась для малых ЭВМ. О росте ее популярности говорит тот факт, что она становится базовой и на супер-ЭВМ. В частности, один из основных производителей супер-ЭВМ фирма «Gray» объявила UNIX стандартом на самой мощной в мире на первую половину 1986 г. супер-ЭВМ «Gray-2» (быстродействие с 64-разрядными словами с плавающей точкой 1.2 млрд/с).

Совместимость операционных систем

При переходе к новой операционной системе желательно иметь совместимость со старой, в которой велась разработка программного обеспечения. Этим уменьшается или исключается переработка уже созданного и отлаженного программного обеспечения.

Совместимость на аппаратном уровне обеспечивается однотипными ЭВМ, использующими одинаковые серии микропроцессора, например, серия ЭВМ ДВК-1, ДВК-2М, ДВК-3, ДВК-4, «Электроника 85». Совместимость имеет место между однотипными моделями ЭВМ, например СМ 3 и СМ 4, СМ 4 и СМ 1420, Д3-28 и Т3-29 и т. д.

Должна существовать совместимость на аппаратном уровне между отечественными ЭВМ и их зарубежными аналогами, но, к сожалению, это не всегда имеет место. Такими аналогами являются: для СМ-1 — HP21-MX; для СМ-2 — HP2112; для СМ-3, Электроники 60, Электроники НЦ 80-20 — PDP 11/20; LSI 11/20; для СМ-4, СМ 1300, Электроники 60 — PDP 11/40; для Искры 1256 — WANG 2200; для ЕС 1840 — PCAT; для СМ 1700 — VAX; для СМ 1800 — PDP 8; Micron-3; для СМ 1638, СМ 50/50/3 (МЕРА 60) — PDP 11; для Искры 1030 — PC AT; для Искры 1140 — PC XT; для Искры 1040 — PC XT (IBM с ОЗУ 1 Мбайт); для ЕС-1850 — PC XT; для ЕС-1860 — PC AT; СМ 1800 (с двумя процессорами) — SYS 86330 A.

При совместимости на аппаратном уровне могут быть использованы одинаковые операционные системы на совместимых ЭВМ.

Сами операционные системы также совершенствуются, и между ними также существует совместимость. Например, операционные системы ДОС,

ФОБОС, РАФОС, ДОС КП или ДОС, ОС РВ, РОСРВ или МС ДОС, РС ДОС совместимы снизу вверх.

Существует совместимость на уровне аналогов. Например, для микро-ЭВМ «Электроника» 60 аналогами операционных систем являются ОС ДВК, ФОДОС (не путать с ФОБОС), РАФОС.

Существует совместимость между отечественными операционными системами и некоторыми зарубежными. Например, имеет место совместимость между РТ-11 и РАФОС; РТ-11 и ОС ДВК; TSX и РАФОС 2; RSX и ОС РВ; UNIX и ИНМОС, ИНМОС-16; МС DOC и АДОС; CP/M и Микрос; SPC, MUMPS и ДИАМС.

В настоящее время большое значение приобретают системы автоматизированного проектирования на базе малых ЭВМ. Эффективность работы на таких системах, в частности время освоения, зависит и от операционных систем.

Литература

1. Norton Peter. MS—DOS und PS—DOS — München. — Wien, Hauser, London, 1985. — 266 p.
2. Готье Р. Руководство по операционной системе UNIX. — М.: Финансы и статистика, 1985.
3. Кристиан К. Введение в операционную систему UNIX. — М.: Финансы и статистика, 1985.
4. Беляков М. И. и др. Инструментальная мобильная операционная система ИНМ ОС. — М.: Финансы и статистика, 1985.
5. Керниган Б., Ритчи Д. Язык программирования Си. — М.: Финансы и статистика, 1985.
6. Бахарев И. А., Горелик М. И. Операционные системы. — М.: Знание, 1986.
7. Хаузер Д., Хирт Дж., Хоукинс Б. Операционная система MSD OS. — М.: Финансы и статистика, 1987.

ОПЕРАЦИОННАЯ СИСТЕМА CP/M

Одной из первых операционных систем для персональных компьютеров была CP/M (CONTROL PROGRAMM FOR MICROCOMPUTER — управляющая программа для микро-ЭВМ). Она разработана в 1975 г. и установлена на многих восьми- и шестнадцатиразрядных ПЭВМ, собранных на микропроцессорах 8080 (отечественный аналог — К580ВМ80), Z80, 8086 (К1810ВМ86). Для этой ОС создано мощное программное обеспечение — трансляторы практически всех языков высокого уровня, текстовые редакторы, системы управления базами данных, игры.

Разумеется, операционная система персонального компьютера должна решать те же задачи, что и любая другая ОС, — задачи управления ресурсами ЭВМ, однако акценты при этом расставляются особым образом. Связано это с характером использования персональной ЭВМ — как правило, один пользователь, интерактивный режим работы и, наконец, очень скромные ресурсы оперативной и внешней памяти. Все эти особенности очень хорошо учитываются в CP/M — система надежна, небольшого размера, производительна, мобильна и проста в использовании. Кроме того, система, как и UNIX, в основном написана на языке высокого уровня PL/M, что способствует ее распространению, помогает переносить с одной ЭВМ на другую.

CP/M состоит из трех основных подсистем — процессора терминальных команд, базовой системы ввода-вывода и базовой системы управления дисками (когда мы говорим о дисках в связи с CP/M, то имеем в виду в основном гибкие диски). При включении компьютера системные файлы загружаются в оперативную память. Под процессором терминальных команд понимается интерпретатор команд языка управления заданиями. С помощью этого языка организуется удобный для человека диалог с машиной, причем к одному из существенных достоинств этого диалога относится возможность его проведения пользователем при очень небольшом объеме знаний о системе.

CP/M содержит шесть встроенных команд:

— DIR — вывод каталога файлов диска;

— REN — переименование файла;

— ERA — уничтожение файла;

— SAVE — сохранение файла на диске;

— TYRE — вывод файла на дисплей или печать;

— USER — выбор области пользователя на диске.

Через процессор консольных команд происходит и обращение к остальным подсистемам. Подсистема работает следующим образом: разбирает командную строку, введенную пользователем с терминала, и вызывает для ее исполнения нужную подсистему. Лучше всего возможности CP/M иллюстрируются набором команд, воспринимаемых процессором консольных команд. Вот примеры некоторых: «оттранслировать программу, хранящуюся в файле», «выдать описание файлов, хранящихся на таком-то диске», «создать файл», «загрузить программу», «переименовать файл», «выдать содержимое файла на консоль», «выдать статус гибких дисков, файлов, устройств», «вызвать CP/M для выполнения команд, записанных в таком-то файле, а не введенных с терминала (что является подобием процедур на языке управления заданиями)» и т. п. Две большие группы команд связаны с редактированием файла и с операциями над файлами. Команды редактирования позволяют вставлять, удалять, соединять строки, заменять, перемещать символы, находить определенную строку и многое другое. В число операций над файлами входит копирование файлов, соединение двух файлов, пересылка файлов с устройства на устройство и т. д. Что касается остальных подсистем, то прямого доступа к ним пользователь не имеет.

Конечно, в такой маленькой ОС, как CP/M, стратегии управления ресурсами выбраны по возможности простые. Например, для управления оперативной памятью применяется описанное нами одиночное непрерывное распределение. Однако даже в такой системе нашлось место для некоторых элементов виртуализации. В CP/M принята концепция логических и физических устройств (терминал, устройство чтения, устройство печати и т. д.), а система в соответствии с указаниями программиста или автоматически связывает логическое устройство с конкретным физическим устройством соответствующего типа.

Для отображения логического уровня данных в физический в файловой системе используются описанные нами выше карты памяти. Стоит обратить внимание на идентификацию файлов в системе. Имя файла состоит из логического имени устройства, собственно имени файла с так называемым расширением, определяющим тип файла, например ASM (ассемблер) или HEX (шестнадцатиричное представление, в котором исполняемая программа загружается в память). Особый интерес представляют файлы типа COM (командные) — это последовательности команд системы, с помощью которых программист получает возможность определить процедуры на языке управления заданиями, т. е. собственные команды. Такая на первый взгляд простая возможность позволяет каждому пользователю создать свой набор команд, в наибольшей степени соответствующий его запросам.

Решения, принятые при создании CP/M, оказались настолько удачными, что на ее основе было создано целое семейство ОС для микрокомпьютеров с 8- и 16-битными словами. Среди них выделяется ОС MP/M, предназначенная для работы в мультипрограммном режиме в реальном времени, а также CP/NET — операционная система для локальных сетей микрокомпьютеров. Вариант CP/M является операционной системой и для распространенного персонального компьютера фирмы IBM.

По функциональным возможностям и стркутуре к операционной системе CP/M-86 близка ОС M86 и микро-DOC. Цифра 86 в этих системах указывает на использование микропроцессора 8086. В ПЭВМ с отличными от 8080 и 8086 микропроцессорами установить CP/M невозможно. M86 обеспечивает на персональных ЭВМ ЕС1840 и ЕС1841 функционирование прикладных программ, систем программирования и программ пользователя.

M86 представляет собой однопользовательскую, однопрограммную операционную систему, проста в применении, диалог с пользователем ведется на русском языке, широко применяется режим меню, имеет развитый набор команд.

В ОС M86 реализованы следующие функции: поддержка работы стандартных внешних устройств; реализация виртуального диска в оперативной памяти объемом до 1,5 Мбайт; поддержка запуска, выполнения и завершения программ, написанных в соответствии с соглашениями M86 и CP/M-86; управление файлами, включающее создание, редактирование, запись, чтение, переименование, уничтожение, защиту, копирование файлов, а также получение справочной информации о них; возможность использования русских букв в именах файлов; связь с ЕС ЭВМ; поддержка интерфейса с операционной системой Альфа-ДОС и т. д.

Резидентная часть операционной системы занимает в основной памяти 28 Кбайт. Остальные программы системы загружаются в память по необходимости. Поставляется на двух дискетах. Общий объем файлов на двух дискетах составляет 364 Кбайт.

Несмотря на доступность и широкое распространение ОС CP/M, она имеет ряд серьезных недостатков, весьма ощутимых для современных пользователей. Во-первых, она была изначально и остается теперь инструментом для опытных программистов, хорошо знакомых со структурой ее команд. Во-вторых, все файлы диска регистрируются в общем каталоге, и если объем диска достаточно велик (винчестер на десяток мегабайт), то управлять файлами весьма трудно.

Более совершенны ОС MS—DOS.

Компьютер в кармане

Американская компьютерная фирма «Atari» продемонстрировала в Париже самый тонкий в мире компьютер, совместимый с персональными компьютерами IBM:

Его размеры — 20×4, 1×3 см, масса — менее 450 г. Компьютер имеет ЖК—дисплей и клавишный пульт, аналогичный формату Qwerty.

Цена компьютера — 470 долл.

New Scientist № 1652. 18.2.89.

Микрокалькуляторы

Фирма «Caiso. Jnc.» производит различные калькуляторы для финансовых и научных расчетов и, в частности, с автоматическим телефонным номеронабирателем.

FC-1000 — графический калькулятор для финансовых расчетов — позволяет пользователю видеть графики на восьмистрочном экране с шестнадцатью столбцами. Калькулятор имеет программируемую память, рассчитанную на 2000 шагов, и 26 независимых ЗУ. Он позволяет производить такие финансовые расчеты, как нахождение простых и сложных процентов, определение накопления, амортизации, обесценивания.

Калькулятор DC-810 с банком данных. Размер его — с кредитную карточку, однако он может хранить 100 номеров телефонов, имен и т. д., а также график работы на год. Калькулятор снабжен рядом сигнальных устройств, секундомером с остановом.

FX-7500G — карманный калькулятор для научных расчетов с графическими возможностями, выполняет 195 функций, снабжен памятью, рассчитанной на 4006 шагов. Информация и формулы располагаются на восьмистрочном экране с 16 символами в строке. Уравнения преобразуются в графический вид одним нажатием клавиши.

QD-350 — калькулятор с восьмицифровым автоматическим номеронабирателем, может хранить 50 наборов данных. Хранимые телефонные номера набираются нажатием клавиши, а последний номер может повторно набираться автоматически. Номера кредитных карточек и телефонов могут быть связаны так, что одним нажатием клавиш производится запись на счетах кредитных карточек.

Office Equipment & Methods
10.88. т.34. с.18.

Настольные рабочие станции

Фирма «Apollo Computer Jnc.» представила две новые серии самых мощных персональных рабочих станций, выполняющих более 7 млн. оп/с. Серии 4500 PSW и 3500 PSW основаны на микросхемах «Motorola MC 68030» — последней версии

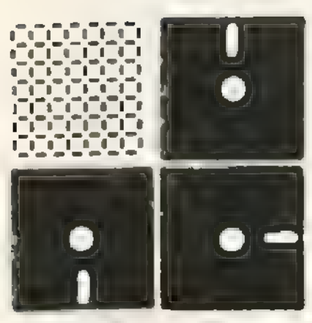
промышленного стандарта микропроцессоров фирмы «Motorola». К числу новинок, входящих в них, относятся: КЭШ с объемом физической памяти 64 кбайта, память с чередованием адресов, а также возможность работы в режиме без ожидания. Это оптимизирует работу станций так, что позволяет максимально использовать весь производственный потенциал.

Office Equipment
& Methods

сентябрь 88 г. т. 34. с. 8.

С. А. УТЕНКОВ

МИКРОПРОЦЕССОРЫ ЭВМ РОБОТОТЕХНИКА



Ю. В. КОЛОСОВ

Контроль влажностных параметров

Задача контроля относительной влажности воздуха является одной из актуальных задач агротехнической практики, связанной с выращиванием овощей в теплицах и парниках, а также при выдерживании и хранении сельскохозяйственных продуктов в овощехранилищах. При этом чаще всего требуется эпизодический контроль с дальнейшим ручным поддерживанием процесса. Однако применение управляющих калькуляторов, особенно с энергонезависимой памятью, может способствовать решению задач автоматического управления влажностными режимами.

Это может быть обеспечено, например, либо за счет подачи влаги в распылители, либо за счет включения вентилятора, осуществляющего обмен воздухом, либо за счет открывания воздухозаборных форточек и т. д. Наконец, может быть организована подача аварийного сигнала при недостаточной (при выращивании в теплице) или избыточной (в овощехранилище) влажности.

Эта же задача возникает и в животноводстве при поддержании и контроле влажностных параметров в местах закрытого содержания скота.

Задача контроля влажности является особенно важной именно в настоящее время, когда повышенное внимание начинает уделяться использованию достижений науки и техники при решении всего комплекса задач агропромышленного комплекса (и в растениеводстве и в животноводстве).

Задача контроля и поддержания относительной влажности воздуха возникает и в других производствах и технологиях, например при эксплуатации сушильного оборудования (сушка фруктов, табака, чая и других продуктов растениеводства, сушка мехов, овчин и т. д.) и, наконец, при поддержании климатических условий в производственных помещениях.

Существуют различные способы контроля влажностных параметров воздуха. В нашем случае наиболее удобным является психометрический метод контроля относительной влажности, использующий измерение температуры сухого и влажного термометра с дальнейшим пересчетом этих измерений в относительную влажность среды. Удобство данного метода заключено в возможности использования ранее описанной методики контроля температуры с помощью двух датчиков: штатных термометров сопротивления и вышепредложенных нормирующих преобразователей. Некоторая дополнительная особенность при данных измерениях состоит в необходимости создания эффекта влажного термометра. Для этого может быть применен чувствительный элемент одного из термометров, завернутый в тонкую ткань (марлю, бязь). Свисающий конец ткани опущен в питающий сосуд с водой (лучше дистиллированной).

Можно предложить конструкцию психрометрического датчика с автоматическим подливом и поддержанием уровня воды в питающем сосуде за счет автомобильного поплавкового регулятора уровня [1].

При определении относительной влажности психрометрическим методом используется полуэмпирическая формула [2, 3]:

$$\varphi = \frac{P_в - A P_{атм} (\theta_с - \theta_в)}{P_с} \cdot 100, \%,$$

где $P_в$ — максимальная упругость пара при температуре мокрого термометра $\theta_в$, мм.рт.ст.

$P_с$ — максимальная упругость при температуре сухого термометра $\theta_с$, мм.рт.ст.

$P_{атм}$ — атмосферное давление, мм.рт.ст.

A — психрометрический коэффициент, 1/град.

$\theta_с$, $\theta_в$ — температуры сухого и мокрого термометров, °С.

Для определения максимальных упругостей пара используется формула:

$$\lg p = 0{,}622 + \frac{7{,}5\theta}{238 + \theta}.$$

Контроль относительной влажности воздуха

Приведена программа, обеспечивающая измерение температур сухого и влажного термометров, пересчет их показаний в относительную влажность среды с дальнейшей индикацией значений влажности, отображаемой в процентах.

Адрес	Команды	Код команды	Содержание команды
01	F3	32	$\theta_в$
02	ПП	68	
03	6	64	
04	F2	22	Перепад температуры $\Delta\theta=\theta_с-\theta_в$
05	В↑	06	
10	F3	32	
11	—	86	
12	7	74	$A=0{,}0007$
13	ВП	66	
14	4	44	
15	/--/	56	
20	X	26	
21	В↑	06	
22	F8	82	Ритм, давление $A\cdot p_{атм}\cdot\Delta\theta$
23	X	26	
24	В↑	06	
25	F7	72	$P_в$
30	↔	16	
31	—	86	$P=P_в-A\cdot P_{атм}\cdot\Delta\theta$
32	P6	61	
33	F2	22	$\theta_с$
34	ПП	68	
35	6	64	
40	F6	62	
41	В↑	06	
42	F7	72	$P_с$
43	÷	36	φ
44	1	14	Нормализация показания φ, %
45	0	04	
50	0	04	
51	X	26	
52	В↑	06	Код повторения эксперимента
53	ВП	66	
54	1	14	
55	0	04	
60	↔	16	
61	С/П	78	
62	БП	58	Возврат в начало программы
63	РО	01	
ПП (подпрограмма)			
64	2	24	238
65	3	34	
70	8	84	
71	+	96	
72	P7	71	
73	↔	16	
74	7	74	7,5
75	·	46	
80	5	54	
81	X	26	
82	В↑	06	
83	F7	72	
84	÷	36	
85	0	04	0,622
90	·	46	
91	6	64	
92	2	24	
93	2	24	
94	+	96	$\lg p$
95	2	24	2,3
100	·	46	
101	3	34	
102	×	26	$\ln p$
103	PI*	33	p
104	P7	71	
105	В/О	48	Возврат из подпрограммы

В приведенной программе все постоянные, используемые в расчете, а именно А-0,0007; 0622; 238; 7,5 и число 2,3, осуществляющее перевод десятичного логарифма в натуральный, введены непосредственно в программу, что несколько (на 18 команд) перегружает программу. Однако при этом упрощается эксплуатация ее, ибо в противном случае пришлось бы вводить эти числа в адресуемые регистры непосредственно перед выполнением программы расчета. Последнее неудобно, так как сопряжено с возможностью внесения дополнительных ошибок при записи чисел в регистры, не говоря уже о том, что количество адресуемых регистров ограничено.

В целом предложенная программа полностью помещается в памяти калькулятора. Остается только ввести в регистр 8 значение атмосферного давления, измеренное в мм. рт. ст. При нормальном давлении вводится число 760.

После набора и контроля программы следует ввести следующие коды: код эксперимента 2100 0001 и регистр 9, обеспечивающий контроль двух параметров и режим повторения эксперимента; затем множители в регистры 2 и 3, обеспечивающие выдачу сигналов из этих регистров в градусах **1 ВП 5** ..... Р2, Р3. Как и ранее, желательно проверить числа, введенные в регистры 8,9,2 и 3. Нажав клавиши **Сх В/О**, подготовить начало процесса. Запустить процесс, нажав кнопку **ПУСК**. При этом на табло высвечивается измеряемая относительная влажность периодически после каждого цикла расчета по программе.

Контроль температуры и относительной влажности воздуха

Часто возникает необходимость контроля и температуры и относительной влажности воздуха.

Ниже приведена программа, обеспечивающая измерение температур сухого и влажного термометров, пересчета их показаний в значение относительной влажности с дальнейшей индикацией температуры сухого термометра — температуры среды и относительной влажности воздуха.

Адрес	Команда	Код команды	Содержание команды
01	F3	32	$^{\theta}$B
02	ПП1	68	
03	7	74	
04	Г2	22	Перепад температуры $\Delta\theta={}^{\theta}c-{}^{\theta}B$
05	В↑	06	
10	Г3	32	
11	—	86	

12	7	74	A=0,0007
13	ВП	66	
14	4	44	
15		56	
20	X	26	
21	B↑	06	
22	F8	82	Ритм, давление
23	X	26	$A P_{атм} \cdot \Delta\theta$
24	B↑	06	
25	F7	72	$P_в$
30	↔	16	
31		86	$P = P_в - A \cdot P_{атм \cdot \Delta\theta}$
32	P6	61	
33	F2	22	^0C
34	ПП1	68	
35	7	74	
40	F6	62	
41	V↑	06	
42	F7	72	
43	÷	36	
44	1	14	Нормализация показаний φ, %
45	0	04	
50	0	04	
51	X	26	
52	ПП2	68	
53	P, ВП	63	
54	F2	22	
55	ПП2	68	
60	P, ВП	63	
61	БП	58	Возврат в начало программы
62	РО	01	
ПП2 (подпрограмма)			
63	B↑	06	Код повторения эксперимента
64	ВП	66	
65	1	14	
70	0	04	
71	↔	16	
72	С/П	78	
73	В/О	48	Возврат из подпрограммы
ПП1 (подпрограмма)			
74	2	24	238
75	3	34	
80	8	84	
81	+	96	
82	P7	71	
83	F4	42	7,5 из регистра 4
84	X	26	
85	B↑	06	
90	F7	72	
91	÷	36	
92	B↑	06	
93	F5	52	0,622 из регистра 5
94	+	96	lg*p*
95	2	24	2,3
100	·	46	
101	3	34	
102	X	26	ln*p*
103	PI*	33	*p*
104	P7	71	
105	В/О	48	Возврат из подпрограммы

В последней программе (более длинной) из нехватки объема программируемой памяти часть постоянных, используемых в расчете, вводится в адресуемые регистры, что, как было сказано, менее удобно, однако это вынужденная мера, определяемая ограничениями данного калькулятора.

В итоге после набора программы следует ввести служебные коды: в регистры 8,9,2 и 3, как и в предыдущей программе. Кроме того, дополнительно числа 7,5 в регистр 4 и 0,622 в регистр 5.

После подготовки процесса и его запуска на табло калькулятора будет периодически высвечиваться относительная влажность в процентах и температура в градусах.

При необходимости организации контроля и управления влажностно-температурным режимом приходится сталкиваться с выше отмеченными трудностями, связанными с ограниченными возможностями применяемых отечественных калькуляторов МК-64. Эти задачи могут быть решены на новом отечественном управляющем калькуляторе типа МК-49, характеристики которого были приведены в предыдущей статье. Серийный выпуск МК-49 еще не начат. Таким образом, можно надеяться, что с появлением следующего поколения управляющих калькуляторов возникнут предпосылки к увеличению их потенциальных возможностей при управлении процессами и технологиями бытового и производственного назначения.

Литература

Психрометр а/с 920492.

Усольцев В. А. Измерение влажности воздуха. — Л.: Гидрометеоиздат, 1959.

Берлинер А. А. Измерение влажности. — Л.: Гидрометеоиздат, 1959.

МОЛОДЕЖНЫЙ КОМПЬЮТЕРНЫЙ ЦЕНТР «КОНТАКТ»

Предлагает желающим прикладные, обучающие и игровые программы для компьютеров БК—0010, КУВТ286, ДВК—2, ДВК—3, Микроша, Специалист, SINKLAIR, ATARI и др.

Наш адрес: 614010, г. Пермь, а/я 4694, тел. 44-01-44.

Генератор идей и исполнитель Стив Джобс (слева) и Джон Скалли. Джобс, основатель и идейный вдохновитель фирмы «Apple Computer, Inc.», начальный успех которой был феноменальным, пригласил опытного бизнесмена Скалли руководить компанией. Сконструированная Джобсом модель «Макинтош» по своим характеристикам была техническим чудом, но не получила широкого распространения в учреждениях. Джобс впоследствии покинул фирму, а компьютер «Макинтош» стал незаменимой принадлежностью «настольных типографий».

Стив Джобс и компания «Apple»

Если бы летом 1976 года вы заглянули в гараж Поля Джобса, что в окрестностях Лос-Альтоса (Калифорния), едва ли вам пришло бы в голову, что здесь, среди хлама с обрезками проводов и исковерканными электронными деталями, рождается предприятие, которое будет ворочать миллиардами долларов. Здесь, в окружении домашней стиральной машины, разбросанных кремниевых микросхем и монтажных плат, 21-летний сын Джобсов Стивен со своим 25-летним приятелем Стивом Возняком трудились над сборкой первых компьютеров «Эппл». Всего пять лет спустя компания «Apple computer» стала живой легендой. Оборот предприятия, начавшего свой путь в невзрачном гараже, к 1981 году достиг 335 миллионов долларов. К 1986 году было продано 2,5 миллиона моделей «Эппл II». Ожидается, что в 1987 году объем сбыта превысит 2 миллиарда долларов.

Предприятие продолжало расти, но, как всякий молодой организм, не могло избежать болезней роста. При всем том «Apple» добилась феноменального успеха. «International Business Machines Corp.» (IBM), крупнейшую в мире компанию по производству ЭВМ, можно считать лидером на рынке персональных компьютеров, однако «Apple» продолжает удивлять технический мир своими новшествами. Как партнеры основатели фирмы «Эппл» Джобс и Возняк не подходили друг другу ни темпераментом, ни стилем работы. Объединял молодых предпринимателей лишь общий интерес к компьютерам.

Джобс, проучившись лишь один семестр в Ридколледже в Орегоне, в 1972 году забросил учебу и нашел место в «Atari inc», одной из ведущих компаний, разрабатывавших первые компьютерные видеоигры.

Возняк, или Воз, как его называли, работал в компьютерной компании «Hewlett—Packard», которая, кстати, тоже когда-то начала с опытов в гараже. Возняк был членом—основателем клуба «Хоумбру компьютер клаб», где небольшая группа техников-энтузиастов собиралась раз в месяц обменяться конструкторскими идеями и замыслами в области компьютеров. И когда другие члены клуба стали сами конструировать компьютеры, вполне естественно, что и наши друзья решили попытать счастья. В результате появился «Эппл I».

Следующий шаг

В 1976 году персональные компьютеры приобретали прежде всего техники-любители, которым доставляло удовольствие возиться с оголенными монтажными платами, какие были в самом первом «Эппл I». Джобс, однако, отлично понимал, что для процветания дела необходимо расширить круг покупателей, и он выделил солидную долю ограниченного капитала предприятия на второй проект Воза — «Эппл II». Для новой модели Джобс заказал изящный пластиковый корпус со встроенной в него клавиатурой. Это была, вне всякого сомнения, удачная идея, благодаря которой компьютер привлек внимание неспециализированного, а значит, более широкого круга потребителей. Впервые персональный компьютер приобрел вид бытового прибора.

Продажа великолепно сконструированной Возняком новинки резко возросла. Люди, никогда не имевшие дела с компьютерами, стали охотно покупать «Эппл II» из-за его «дружелюбного» вида. Под пластиковым корпусом Возняк разместил исключительно эффективную электронно-вычислительную машину. Она не только отвечала последнему слову микрокомпьютерной техники, но и несла на себе печать индивидуальности конструктора. Компьютер как бы приглашал потребителя вникнуть в его сокровенную суть. В отличие от конкурирующих моделей «Эппл II» имел так называемую открытую систему. Его технические данные и спецификация позволяли потребителю разрабатывать собственные программы и добавочные устройства для расширения сферы применения компьютера. Это обстоятельство еще больше способствовало сбыту новой модели, и в течение года возникли сотни компаний по составлению программ и изготовлению дополнительных блоков памяти и печатающих устройств, расширяющих возможности компьютера. «Эппл II» стал пионером мощной электронной индустрии.

Опасность успеха

Но с успехом пришли и волнения. Все ждали, что в течение двух лет появится новая модель. Первые шаги с новым компьютером — «Эппл III», о выпуске которого компания объявила, не дождавшись окончательной технической доводки модели, оказались неудачными. В довершение ко всем неприятностям на рынке появился новый конкурент. Это была IBM, крупнейшая в мире компьютерная компания, своими размерами превосходящая «Apple» по крайней мере в 130 раз.

Не прошло и года, как персональные компьютеры IBM свели на нет первенство машин «Эппл» на рынке и почти лишили калифорнийскую компанию доступа к самым перспективным покупателям персональных компьютеров — коммерческим предприятиям.

Индустрия персональных компьютеров признала своим лидером IBM. Появились сотни фирм, которые выпускали ЭВМ, похожие на персональные компьютеры IBM или работавшие в сочетании с ними. Но Джобс не думал сдаваться. «Apple», строившая машины, технологически отличавшиеся от продукции IBM, была полна решимости постоять за себя. Ответом конкуренту явилась «Лиза» — самый дорогой компьютер фирмы «Apple».

Идея новой машины была заимствована у совершенно посторонней компании — гигантской корпорации копировальных машин «Ксерокс». В начале 80-х годов Джобс побывал в центре компьютерных разработок фирмы «Ксерокс» в «Силиконовой долине». Он был ошеломлен тем, что увидел. Здесь впервые использовали указательное устройство под названием «мышь» размером с пачку сигарет. «Мышь», перемещаемая рукой по столу, в свою очередь, передвигает стрелку на экране, на котором изображены графические символы, представляющие выполняемые компьютером функции.

Джобс увидел в новшестве «Ксерокса» средство сделать компьютер массовым аппаратом. Однако возникли трудности, когда стоимость разработки компьютера «Лиза» стала выходить из-под контроля. Продажная цена угрожала повыситься до 10 000 долларов, в то время как «Эппл II» стоил 2500 долларов. Джобс не мог на это пойти. Он понимал, что компания нарушает собственный девиз: «Каждому по компьютеру». Мало кто мог себе позволить платить 10 000 за компьютер. И после яростных споров с президентом фирмы Джобс, молодой председатель совета директоров, оказался в изоляции. Его отстранили от руководства проектом «Лиза», и бразды правления были переданы менеджерам с большим опытом работы в крупных компьютерных компаниях Соединенных Штатов.

Появление «Макинтоша»

Джобс не стал дорабатывать «Лизу» на благо компании, а взялся за новый независимый проект, который в конечном счете сыграл главную роль в провале «Лизы». Однако для фирмы важнее было то, что Джобс проложил дорогу к удачной модели «Макинтош».

Джобс организовал элитарную группу для работы над недорогим уменьшенным вариантом «Лизы» под названием «Макинтош» (наименование одного из сортов американских яблок). Группа обосновалась в стороне от главного здания фирмы, и началась трехгодичная, без сна и отдыха,

гонка за разработку «самого умопомрачительного компьютера», как часто любил говорить о нем Джобс.

В основных чертах «Макинтош» выглядел как младший брат «Лизы». У него были такой же высококачественный экран для графических изображений и приспособление «мышь». Самым трудным было скомпоновать персональный компьютер стоимостью 2500 долларов. Конструкторам это удалось. Молодые энтузиасты из бригады «Мак» трудились самоотверженно и ценой больших усилий добились успеха. Они были горды этим своим детищем, но соперничество между проектами «Лиза» и «Мак» отразилось на морали коллектива. Отразилось это и на продаже «Лизы».

Переходный период и снова взлет

Тогда в руководстве фирмы произошли большие перемены. После кропотливых поисков Джобс утвердил президентом фирмы Джона Скалли, 44-летнего президента компании «Пепсико», производящей пепси-колу.

В январе 1984 года «Макинтош» поступил в продажу. Удобный и «симпатичный» компьютер обратил на себя внимание прессы и общества. Это был вызов фирмы «Apple» всемогущей IBM, противостояние молодости и идеализма корпоративной мощи и размаху. Старт модели «Макинтош» оказался весьма впечатляющим. Стив Джобс снова был на коне или по крайней мере в то время так казалось.

После произведенной Скалли перетряски дела фирмы существенно улучшились. С появлением более мощных моделей уникального «Макинтоша» сбыт пошел вверх. Другим толчком к оживлению бизнеса послужил неожиданный и стремительный расцвет небольших независимых издательств, использующих графические возможности последних моделей персональных компьютеров. В сочетании с новейшими лазерными печатающими устройствами «Макинтош» со своим набором великолепных и удобных в работе графических программ оказался идеальной находкой для индивидуальных или небольших «настольных» издательств. Джобс совершенно правильно предугадал перспективность простых персональных компьютеров, но не успел вовремя доработать все детали.

Хотя «Apple» существует всего десять с лишним лет, компания крепко стоит на ногах как на рынке сбыта, так и в сфере производства. Теперь «Apple» упоминается в одном ряду с IBM как солидная ведущая корпорация. Ее продукция продается в 85 странах мира. Будущее компании зависит от того, сможет ли она по-прежнему поставлять на рынок новинки без Джобса с его даром технического предвидения. В любом случае компания «Apple Computer, Inc.» навсегда войдет в анналы истории американского бизнеса как наглядный пример того, чего можно достичь при сочетании благоприятных условий, новаторства, упорного труда, удачи и решимости.

Новая модель фирмы «Эппл»

В сентябре 1986 года фирма «Эппл» выпустила модель «Эппл II С» — новейшую модификацию из прославленной серии персональных компьютеров «Эппл II». Новинка работает со всеми существующими программами «Эппл» и сверх того имеет немало собственных новых программ. Лицам, интересующимся графическим дизайном, предлагается на выбор 4000 различных цветов и оттенков, причем одновременно на экране можно оперировать 64 различными цветами. Новая модель снабжена 15-канальной звуковой системой, способной дублировать слова, появляющиеся на экране, или воспроизводить музыкальные тоны. Важное преимущество этой модели — большая скорость; она прогоняет программы в три раза быстрее, чем родственные варианты «Эппл II», «Эппл Пе» и «Эппл Пс». Руководители компании надеются, что новая модель с ее мощным микропроцессором и богатейшей гаммой цветов и звуков создаст новое поколение покупателей персональных компьютеров, как это уже сделала модель «Эппл II» десять с лишним лет назад.

ПРОИЗВОДСТВЕННЫЙ КООПЕРАТИВ «РАДАР»

МАКСИМУМ ЭФФЕКТА И МИНИМУМ ЗАТРАТ для пользователя — такой цели достигли производственная фирма «РАДАР» и центр рефлексотерапии, разрабатывая «АРМ РЕФЛЕКСОТЕРАПЕВТА»

Комплекс позволяет проводить экспресс-диагностику по методу Акабанэ, бытро завоевавшему популярность у рефлексотерапевтов страны. Инфракрасный излучатель, предназначенный для воздействия на точки акупунктуры, подключается к персональной ЭВМ через интерфейсный блок, в котором зашиты управляющие программы, позволяющие дозировать ИК излучение и последовательно фиксировать показания реакции обследуемого на болевой порог. Далее компьютер производит обсчет для акупунктурного воздействия. Комплекс успешно используется во многих клиниках страны и экспонирован на ВДНХ СССР. Стоимость комплекса — 4790 руб.

АРМ рефлексотерапевта необходим и доступен каждому терапевту!

Расширить функциональные возможности, повысить быстродействие, продлить срок службы ваших компьютеров позволит электронный квазидиск.

Электронный квазидиск выполнен на базе СБИС ОЗУ 41256 (фирмы Сименс). Объем памяти — 512 кбайт. Предназначен для работы в составе ПЭВМ типа ДВК, УКНЦ. Стоимость значительно ниже соответствующих аналогов: 1998 руб. В комплекте — диск НГМД с драйвером.

Адрес: 428034, г. Чебоксары, Московский просп., 40. Приборостроительный завод, «Радар», тел. 24-51-75.

UNIX — ПО СТРАНИЦАМ ЖУРНАЛА «ЭЛЕКТРОНИКА»

Операционная система UNIX реального времени. В 1989 г. следует ожидать появления операционной системы UNIX реального времени с гарантированным временем реакции от 50 до 100 мкс. Разработчикам корпорации Concurrent Computer Corp. (Уэстфорд, шт. Массачусетс), образованной в результате слияния фирмы Masscomp Computer Corp. и компании Concurrent Computer (Тинтон-Фолс, шт. Нью-Джерси), удается достичь столь малого времени реакции благодаря реализации многопроцессорной архитектуры на базе RISC-процессоров с быстродействием 25 млн. команд в секунду и нескольким новым оригинальным программным решениям. Микропроцессорный вычислительный комплекс будет содержать до восьми ЦП, каждый из которых строится на базе комплектов кристаллов корпорации MIPS Computer Corp. (Саннивейл, шт. Калифорния).

Графические средства сопряжения ОС UNIX и MS-DOS. Благодаря графическому интерфейсу пользователя фирмы Hewlett-Packard, получившему название Common X Interfase (CXI), системы, работающие под управлением ОС UNIX, могут теперь функционировать аналогично персональным компьютерам, работающим под управлением программных средств Microsoft Windows и Presentation Manager фирмы Microsoft и в рамках ОС MS-DOS и OS/2 соответственно.

CXI дает разработчикам программного обеспечения общие принципы и средства для создания прикладных программ на базе единого интерфейса пользователя. В результате пользователи, знакомые с программными средствами Microsoft Windows и Presentation Manager, могут перейти на ОС UNIX с минимальными усилиями на переподготовку.

Основа CXI — это система X Window System, которая является действующим стандартом ОС UNIX. Такой общий интерфейс, связывающий различные ЭВМ и информационные системы, состоит из трех основных элементов: модуля «стиля программирования» фирмы Microsoft, интерфейса прикладных программ X Widget фирмы HP и средства Window Manager (менеджер полиэкранного режима) фирмы HP.

В настоящее время существуют две версии CXI, одна из которых реализует двумерную форму представления данных на экране. Кроме того, фирма HP предлагает версию CXI с трехмерным представлением информации, чтобы, например, при выборе и «нажатии» на изображенную на экране клавишу она выглядела нажатой.

Блок «стиля программирования» и программный интерфейс систем CXI поставляется фирмой HP разработчикам программного обеспечения и систем в составе пакета X Windows версии 11 по цене 710 долл. Средство по управлению окнами экрана, которое нужно для разработки разных прикладных программ, появится в 1989 г.

О едином стандарте ОС UNIX. В начале 1989 г. стала реальной возможность объединения всех разновидностей ОС UNIX в единый стандарт. Однако весьма маловероятно, к большому сожалению, что этот стандарт будет иметь единственный вариант. Дело в том, что свои права на собственную реализацию стандарта ОС UNIX настойчиво предъявляют Open Software Foundation (Фонд открытых программ) и группа компаний — приверженцев версии ОС UNIX, предложенной компанией AT&T. В процессе эволюции ОС UNIX, разработанной в компании Bell Laboratories корпорации AT&T еще в 1969 г., появилось множество ее версий. И неудивительно, что между организациями и компаниями возникли разногласия в отношении того, какая из версий ОС UNIX должна стать отраслевым стандартом. Но независимо от того, какая версия ОС будет выбрана, все прикладное программное обеспечение должно оставаться совместимым на уровне исходного кода.

Надежды сторонников ОС UNIX на достижение единства основываются главным образом на возможности соглашения по вопросу технических требований стандарта, подготовленных и сформулированных компанией X/Open Co. Ltd. (Лондон и Сан-Франциско).

Если предожение организации X/Open будет принято, то это в немалой степени связано с предпринимавшимися на протяжении некоторого времени усилиями. Один из важных новых стандартов, Posix (Portable Operating System Interfase for Computer Environments — мобильная операционная система для вычислительных сред), был принят в 1988 г. институтом инженеров по электротехнике и радиоэлектронике. X/Open и другие организации заняты разработкой важных дополнений, не вошедших в стандарт Posix.

Соперничество между различными версиями ОС UNIX прекратится, если организации X/Open удастся стать признанным единственным автором стандарта на ОС UNIX. Возможность такого исхода предвидят многие наблюдатели. «Я полагаю, что в конце концов будет существовать один компромиссный стандарт — Posix (предложенный комитетом ИИЭР по стандартизации ОС UNIX), а также соответствующие предложения правительственных учреждений США и организации X/Open сольются воедино», — говорит Расс Олдридж, руководитель маркетинга в отделении работающих под управлением ОС UNIX вычислительных систем фирмы Convergent Technologies Inc.

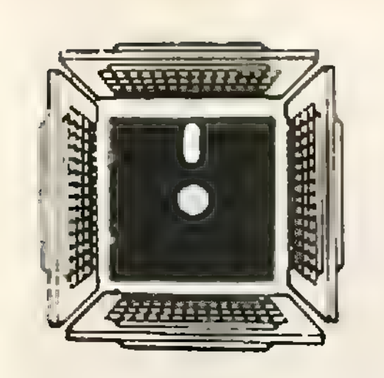
Сейчас наибольший интерес вызывают такие версии UNIX: UNIX System V.4 компании AT&T; 3 ОС AIX корпорации IBM; OSF-1 и некоторые другие.

Вопрос о том, какая из версий ОС UNIX получит наибольшее распространение, будет решен на рынке. Но так ли существенно для пользователя, кто окажется победителем в этой битве?

Согласно прогнозам компании Infocorp (Маунтин-Вью, шт. Калифорния), занимающейся исследованием рыночной конъюнктуры, мировой объем рынка систем, базирующихся на ОС UNIX, вырастет с 21,6 млрд. долл. в 1990 г. до 34,4 млрд. долл. в 1993 г. По оценкам этой же компании, в 1988 г. объем этого рынка составил 9,8 млрд. долл.

Важнейший результат всех этих споров состоит в том, что независимо от того, какая версия будет преобладать в отрасли — или явного лидера среди них не окажется вовсе, — конечные пользователи в любом случае окажутся в выигрыше. Версии OSF-1 и V.4 будут совместимы на уровне исходного кода. Пользователям, разрабатывающим собственные программы, будет достаточно создать всего лишь один вариант программного изделия, который можно будет исполнять на вычислительных системах различных изготовителей. Если же они приобретут систему, работающую под управлением другой версии ОС UNIX, то все, что от них понадобится, — это перекомпилировать исходный код.

Существует мнение, будто бы ОС UNIX не обеспечивает достаточно надежную защиту вычислительных систем; однако при более внимательном изучении в ОС UNIX можно обнаружить большое число различных механизмов защиты. Последние, однако, оказались скрытыми от пользователей из-за плохой документации. Чтобы исправить такое положение, рабочая группа по защите данных организации X/Open разработала руководство по обеспечению безопасности вычислительных систем (см. Электроника, 1989, № 3).



С Новым годом!
С новой рубрикой!
Открывая новый раздел, редакция надеется, что он будет регулярным, массовым, самое главное — интересным и полезным. Вести его будет уже известный вам Б. А. Тарасенко, лектор общества «Знание», научный сотрудник НПО «Энергия». Автор «Алгоритмического букваря» (ВТ 4-88).

Б. А. ТАРАСЕНКО

МЫ И ПАПА
Выпуск 1

Если обратиться к заголовку, то кто это мы? Мы — это будущие и настоящие авторы рубрики, ее, надеемся, активные и доброжелательные читатели, готовые учиться и учить занимательному, интересному, новому. Это те, кому можно сказать, например, так:

Твори, гори, но не сгорая,
Умей, дерзай, но меру знай.
Коль вдруг зимой захочешь мая,
Пусть будет впрямь душевный май!
Всем нужно новое мышленье,
Труд в постиженье красоты.
Мечты восторг и вдохновенье
Не прилетят из пустоты!
Не сотвори себе кумира,
Его лишь одного любя,
На радость всем и в пользу мира
Ты переделывай себя!

Сейчас еще рано, а через некоторое время хотелось бы словами А. С. Пушкина констатировать: «Друзья мои, прекрасен наш союз!» — и взять их нашим постоянным эпиграфом.

Папа у каждого свой, единственный, а у нас в рубрике будет один, общий. И конечно же, еще персональный, домашний. Без него наши материалы лучше не читать. Потому что ПАПА — это Простейший Алгоритмический Программируемый Автомат. Сокращением ПАПА, заглавными буквами, но без кавычек, в дальнейшем будем именовать семейство программируемых калькуляторов типа «Электроника МК-61», а модели этого семейства писать, в свою очередь, для краткости без кавычек и без слова «Электроника». Вот они, наши программно-совместимые «рабочие лошадки»: БЗ-34, МК-54, МК-61, МК-52, МК-56. Есть и «двоюродные братья»: БЗ-21, МК-46, МК-64. Это все устройства с так называемым клавишным языком. На подходе «вундеркинды» МК-85 и МК-90 (ПК-100) с языком БЕЙСИК. Знаем и помним о зарубежных моделях карманных ЭВМ, которые можно отнести к классу ПАПА. Рассказывать намереваемся о всех. Но в первую очередь о тех отечественных, которые не сняты с производства, имеются «на прилавке» и по доступной цене.

Область маломерной вычислительной техники нуждается в новых названиях и терминологии, необходимы более гибкие и универсальные системы нотации программных текстов, путеводная нить в лабиринтах калькуляторной литературы, развитие хорошего потребительского вкуса: программа программе рознь! Почему бы не заняться этим нам с вами, читатель?

Изготовители семейства ПАПА МК-61 пытаются повернуться «лицом к человеку», к потребителю, но делают это медленно, неумело. Похоже, не очень знают, а что ему, потребителю-то, надо? Поможем? При вашем согласии и участии мы создадим постоянно действующее коллективное техническое задание на новые модели ПАПА и на программное обеспечение к ним.

В наших планах сотрудничества с вами не исключается программирование на конкурсной основе, несложные электронные самоделки, повышающие эксплуатационные параметры конкретных моделей ПАПА.

По количеству проданных в личное пользование единиц и по низкой розничной цене семейство МК-61 далеко опережает в развитии остальной отечественный парк персональных ЭВМ. По глубокому убеждению автора, ПАПА остаются единственной возможностью на ближайшие несколько лет в создании массовой компьютерной грамотности нашей страны. Для успеха дела подобные учебные ЭВМ должны быть и дома, и в аудиториях учебных заведений. Дома ПАПА уже есть, есть реальная возможность увеличения их количества в личной собственности. Их практически нет в школах и на производстве. Но положение поправимо. И

новая рубрика в этом сможет помочь, в частности, соответствующей агитацией.

Все начинается с простого. Нет ничего проще для первого выпуска рубрики, чем простые числа. Правда, Валерий Брюсов напоминал, что «простые числа все не разгаданы». Однако простота от определенной доли таинственности только выигрывает. Напомним, что простым называется число, большее единицы, которое делится только само на себя и на единицу. Эти числа остаются постоянным объектом внимания профессионалов и любителей. Сегодня мы предлагаем вам циклический программный демонстратор первых 25 простых чисел, укладывающихся в первой сотне. В свое время в городе Ессентуки он на целых 10 минут привлек внимание девочки Светы, третьеклассницы из-под Ленинграда, которая, впервые узнав, что это такое, тут же половину простых чисел запомнила. Пригодна модель МК-61 и все семейство этих ЭВМ.

Программа № ПАПА-1/90/1

00	ВСТЕК	ОЕ	49	В/О	52
01	2	02	50	2	02
02	6	06	51	9	09
03	+−+	14	52	В/О	52
04	−	11	53	3	03
05	БП	51	54	1	01
06	0 9	09	55	В/О	52
07	2	02	56	3	03
08	5	05	57	7	07
09	Х−+П О	40	58	В/О	52
10	9	09	59	4	04
11	8	08	60	1	01
12	П−+Х О	60	61	В/О	52
13	3	03	62	4	04
14	×	12	63	3	03
15	−	11	64	В/О	52
16	Х−+П 9	49	65	4	04
17	К ПП 9	−9	66	7	07
18	С/П	50	67	В/О	52
19	ФЛО	5Г	68	5	05
20	1 0	10	69	3	03
21	БП	51	70	В/О	52
22	0 7	07	71	5	05
23	0	00	72	9	09
24	2	02	73	В/О	52
25	В/О	52	74	6	06
26	0	00	75	1	01
27	3	03	76	В/О	52
28	В/О	52	77	6	06
29	0	00	78	7	07
30	5	05	79	В/О	52
31	В/О	52	80	7	07
32	0	00	81	1	01
33	7	07	82	В/О	52
34	В/О	52	83	7	07
35	1	01	84	3	03
36	1	01	85	В/О	52
37	В/О	52	86	7	07
38	1	01	87	9	09
39	3	03	88	В/О	52
40	В/О	52	89	8	08
41	1	01	90	3	03
42	7	07	91	В/О	52
43	В/О	52	92	8	08
44	1	01	93	9	09
45	9	09	94	В/О	52
46	В/О	52	95	9	09
47	2	02	96	7	07
48	3	03	97	В/О	52

Слева от текста программы поставлен ее номер, в котором угадывается ее принадлежность к рубрике ПАПА, указан номер текущего выпуска ВТ, 1/90, последним проставлен порядковый номер сквозной нумерации программ, размещаемых в рублике.

В программе проведены замены обозначений клавишных команд по сравнению с изображениями на клавиатуре ПАПА. На шаге 00 обозначение ВСТЕК заменяет букву В с вертикальной стрелкой справа. На шаге 03 комбинация +−+ заменяет горизонтальную обоюдоострую стрелку. В шагах 09, 12 и 16 комбинация −+ заменяет горизонтальную стрелку вправо, а строчные буквы «х» заменены на заглавные. На шаге 19 русские буквы Ф и Л соответственно заменяют созвучные латинские. Слева — столбец двухразрядных десятичных цифр. Это шаги программы, или адреса ее команд. Справа — двухзначные комбинации индикаторных кодов, такие, какими они появляются на индикаторе ПАПА МК-61, например, в процессе набора текста программы. Надеемся, что после сделанных разъяснений трудностей с набором программы не осталось.

Программа после ее набора запускается на счет просто. Достаточно нажать В/0, набрать целое число в диапазоне от 1 до 25 (порядковый номер простого числа, которое вы желаете увидеть) и нажать С/П. После нескольких секунд мигания индикатора происходит останов счета и высвечивается простое число, которое вы вызывали по порядковому номеру. При дальнейших нажатиях на С/П вслед за относительно непродолжительными миганиями наш «электронный исполнитель» будет

выдавать по очереди простые числа не более 100 в возрастающем порядке. После прохода через 97, последнее простое число в первой сотне, программа автоматически переходит на первое простое число 2 и снова продолжает выдавать числа в нарастающем порядке. Для вызова произвольного числа из 25 возможных необходимо после останова счета нажать В/0 и вслед за этим порядковый номер вызываемого числа. Затем, разумеется, нужно снова нажать С/П.

Вдумчивого читателя всегда интересуют не только внешние эффекты от работы программы, но и то, как они обеспечиваются, какие для этого использованы приемы программирования. В этом смысле с рассматриваемой программой тоже все просто. Основную ее часть составляют 25 подпрограмм, содержащих заготовки двухразрядных простых чисел, причем одноразрядные числа 2, 3, 5 и 7 дополнены слева незначащими нулями. Каждая подпрограмма заканчивается командой возврата в основную программу, то есть В/О, занимает по 3 шага каждая, а в целом зона подпрограмм занимает 75 шагов от 23 до 97 включительно. Чтобы осуществить обращение, нужно задать из основной части программы начальный адрес подпрограммы. Это делает команда К ПП 9 на шаге 17. Подпрограмма ПП — это корневая часть названия команды, напоминающая о ее назначении. К — признак косвенной адресации. Регистр 9, или Рг9, — это место, где хранится адрес обращения к подпрограмме. Содержимое Рг9, или сРг9, — это и есть адрес обращения. В шагах с 10 по 20 организован одинарный простой цикл на базе регистра циклового назначения Рг0. Собственно команда цикла расположена в шагах 19 и 20. Ее клавишная последовательность есть Ф ЛО 1 0 Мы специально здесь не ставим точку и другие знаки препинания (да простит нас корректор!), чтобы не перепутать точку печатную с точкой клавишной. При каждом прохождении цикла сРг9 меняется. Команда на шаге 17 заключена внутри цикла и осуществляет обращение к очередной подпрограмме. Возврат из подпрограммы вместе с «захваченным» оттуда числом происходит на шаг 18, или на команду С/П, или на команду стоп-индикации. В момент передачи управления счетом на эту команду мы и наблюдаем очередное простое число после останова счета. В шагах 19 и 20 параметр цикла, сРг0, автоматически уменьшается на единицу. На шаге 12 сРг0 извлекается из Рг0, на шаге 13 и 14 домножается на 3 (длина каждой из подпрограмм равна 3 шагам), на шаге 15 эта величина вычитается из константы 98 (константа 98 «лежит» в шагах с 10 по 11) или из общего объема программной памяти (в шагах), на шаге 16 результат заносится в Рг9. В шагах с 00 по 06, а также 09 происходит формирование начального значения параметра цикла в соответствии с введенным порядковым номером простого числа.

Что еще поведать о программе? Осторожно выражаясь, оформление числовых констант в виде подпрограмм встречается как прием программирования для простейших ЭВМ нечасто, если не впервые. В остальном программа скажет сама за себя.

В прошедшем году вышли книги С. И. Шапиро и А. Б. Бойко «Программируемые микрокалькуляторы в обучении» и учебник для средних специальных учебных заведений «Микрокалькуляторы в курсе математики», авторы Е. Д. Куланин, Н. Н. Лемешко и В. Л. Шамшурин. Первая отличается разнообразием тем, приемов и стилей программирования. Здесь легко найти полезное и для школы (но далеко не все; например, не все игры годятся для этого), и для любительского досуга дома. Вторая книга выдержана в более строгом стиле, возможно, что это вообще первый учебник, который поддерживает курс математики и информатики на программируемых калькуляторах. Особенно приятно отметить, что в обоих случаях нотация программ почти такая же, какой мы будем придерживаться в нашей рубрике.

В дальнейшем мы постараемся напоминать каждый раз о существовании хотя бы пары старых и новых книг о ПАПА и программировании на нем, сопровождая напоминание хотя бы краткой характеристикой книг.

Что можно сказать о терминологии в

первом выпуске рубрики, даже у́же — о названии нашего вычислительного средства? Если иметь в виду семейство МК-61 и его конструктивное, системное, решение, то наиболее уместно понятие гибрида компьютера (простейшего!) с инженерным калькулятором, дадим ему название компьютер-калькулятор, сокращенно КОК. Когда же в нашей рубрике равноправным участником появится МК-85, а мы верим, что это будет скоро, и КОК и КПК (карманный программируемый калькулятор) сильно потускнеют и постареют как названия (но не как устройства!). Так что с ПАПА́ (давайте договоримся произносить это новое и необычное для нас слово на французский манер, с ударением на последний слог) надежнее. ПМК уже не годится на сегодня и на ближайшее будущее как устаревшее и неточно отражающее суть дела название. Наверно, достаточно будет напомнить, что в недавно вышедших в свет словаре иностранных слов и толковом словаре терминов по вычислительной технике слова-понятия «микрокалькулятор» нет вообще! Тем более напрашивающегося слова-антипода «макрокалькулятор». Нет программируемого микрокалькулятора.

ПМК — это сокращение из разряда слов-однодневок. Как в свое время слово «транзистор» для обозначения транзисторного радиоприемника, они стремительно взлетают у всех на виду, а потом медленно, но верно опускаются в небытие. Это словесный мусор, и убрать его — святое дело. К тому же современному плюрализму мнений должен соответствовать плюрализм названий.

От «микро» в «микрокалькуляторе» можно избавиться легко. Действительно, так ли уж важно, что из микросхем или макросхем сделаны электронные «счеты», занимают ли они добрых полстола или свободно умещаются в кармане пиджака?

ПМК, КПК, КОК. Кто из них «приживется»? Кто — нет? Сказать трудно. Будут и новые «имена». Но вот от слова «калькулятор», входящего в эти сокращения, нам, видимо, никогда не избавиться. Так давайте договоримся, какой смысл сегодня мы будем вкладывать в это слово, в это понятие, а заодно вспомним о понятиях «программируемый калькулятор» и «компьютер».

Под калькулятором будем понимать вычислительное устройство, способное вручную выполнять поодиночке жестко запрограммированные на заводе-изготовителе клавишные операции, но не способное объединять их с запоминанием в последовательность и автоматически выполнять ее. Носителем и исполнителем произвольного алгоритма в этом случае является человек-оператор.

Программируемый калькулятор (он же ПМК, КПК, КОК), или разновидность простейшей ЭВМ, способен объединять отдельные клавишные команды в последовательную цепочку с ветвлениями, запоминать ее и потом автоматически исполнять после подачи единственной стартовой ручной команды. Никаких преобразований команд перед исполнением командной последовательной цепочки не происходит. Это та же самая последовательность действий, которую вы можете выполнять вручную. Разница в том, что последовательность команд, записанную в КОК, можно повторить произвольное число раз с высокой против ручного режима степенью надежности и, как правило, с увеличенной против ручного режима скоростью. При этом на индикатор выдаются специальные сообщения только при самых грубых некорректностях, например при попытке деления на нуль.

Под компьютером будем понимать устройство, способное не только объединять в цепочку и запоминать последовательность ручных команд, а также автоматически исполнять последовательность команд, но и транслировать (переводить) перед исполнением алгоритма его запись в исходном языке, более удобную для человека, в запись на машинном языке, точно соответствующую особенностям компьютера каждого конкретного типа. Запись в машинном языке можно истолковать как порядок нажатия большого числа воображаемых кнопок «большого программируемого калькулятора». Разница между КОК и компьютером в том, что у компьютера побольше «кнопок», а

тем более их допустимых комбинаций, которые практически невозможно запомнить и применять осмысленно. Степень сложности для запоминания и осмысления можно сравнить разве только с несколькими сотнями и даже тысячами китайских иероглифов. При этом нужно отметить, что исходный язык по отношению к разным компьютерам, настроенным на него, является универсальным, одним и тем же, с точки зрения человека. Это очень удобно. Транслятор (переводчик) с этого универсального языка (специальная программа) в машинный код, как правило, способен выдавать внушительный перечень сообщений о некорректностях в записях на исходном языке.

Несмотря на существенную разницу, по нашим определениям, между компьютером и программируемым калькулятором, мы объединяем КОК с классом компьютеров, напоминая каждый раз, когда это необходимо, что это простейший компьютер, или простейшая ЭВМ. В наших рассуждениях возможны неточности, но они второго порядка. Мы просим читателей обратить внимание на главное и двигаться дальше, понимая, что без этих определений движение вперед было бы практически невозможно.

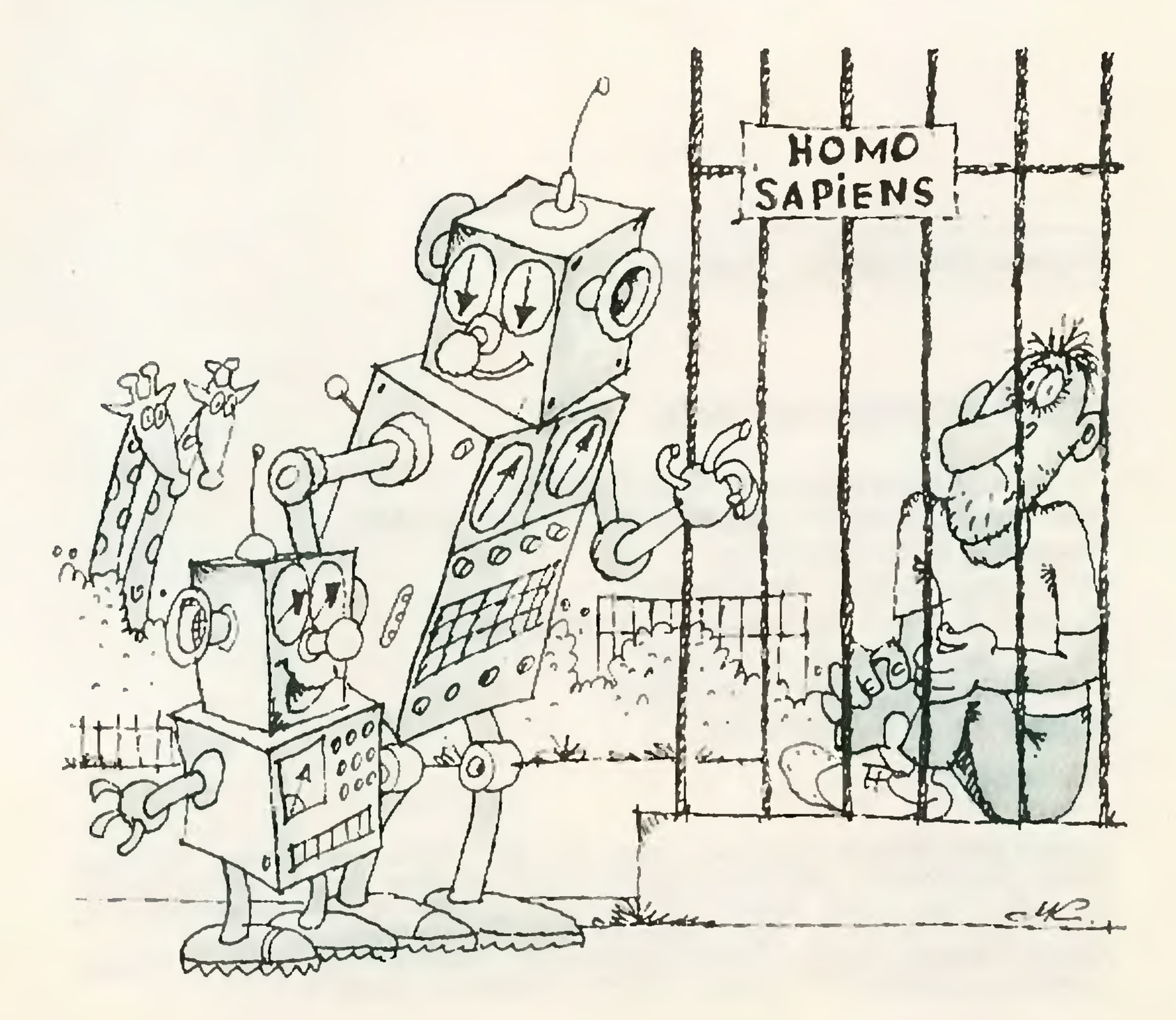
В следующем выпуске рубрики мы займемся элементами словесного диалога с ПАПА на базе программы, вычисляющей корни квадратного уравнения, а также учебным алгоритмическим языком (УАЯ); другие сокращения, используемые в учебных заведениях, — это РАЯ (референтный), ШБАЯ (школьный безмашинный) и другие применительно к ПАПА.

В заключение первого выпуска рубрики Мы И ПАПА предлагаем вам простой ручной алгоритмический этюд для МК-61 и МК-52.

Выполните последовательные клавишные операции, предварительно включив питание, так:

8 3 1 К ИНВ БП 2 +−++−+

Здесь каждая слитная символьная группа или отдельно стоящий символ, разделенные пробелами, обозначают конкретную клавишу ПАПА МК-61 или надклавишную надпись. При правильной работе на индикаторе появится надпись ВСЕ. Но это далеко не все, о чем мы с вами хотели бы еще много раз побеседовать. Пишите нам.



Операционные системы. — М.: Знание, 1990. — 48 с.
0-60 — (Новое в жизни, науке, технике. Сер. «Вычислительная техника и ее применение»; № 1).
ISBN 5-07-000336-4
20 к.

Для чего нужна операционная система? Каков ее состав и возможности? Как работать с операционной системой? Каковы особенности наиболее популярных операционных систем? Ответы на эти вопросы даны в брошюре, рассчитанной на широкий круг читателей.

4040000000 **ББК 32.973**

Научно-популярное издание

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Гл. отраслевой редактор *Л. А. Ерлыкин*
Зам. гл. отраслевого редактора *Г. Г. Карвовский*
Редактор *Б. М. Васильев*
Мл. редактор *Н. А. Васильева*
Художники *В. Н. Конюхов* и *К. Н. Мошкин*
Худож. редактор *М. А. Гусева*
Техн. редактор *А. М. Красавина*
Корректор *В. В. Каночкина*

ИБ № 9807

Сдано в набор 22.09.89. Подписано к печати 16.11.89. Т-18612. Формат бумаги 70×100 1/16. Бумага офсетная № 2. Гарнитура гельветика. Печать офсетная. Усл. печ. л. 3,90. Усл. кр.-отт. 8,45. Уч.-изд л. 4,32. Тираж 70 116 экз. Заказ 598. Цена 20 коп. Издательство «Знание». 101835, ГСП, Москва, Центр, проезд Серова, д. 4. Индекс заказа 904701.
Ордена Трудового Красного Знамени Калининский полиграфический комбинат Государственного комитета СССР по печати. 170024, г. Калинин, пр. Ленина, 5.

Цена 20 коп.

Индекс 70195

Адрес подписчика:

Сол. 5-27

Издательство «Знание»

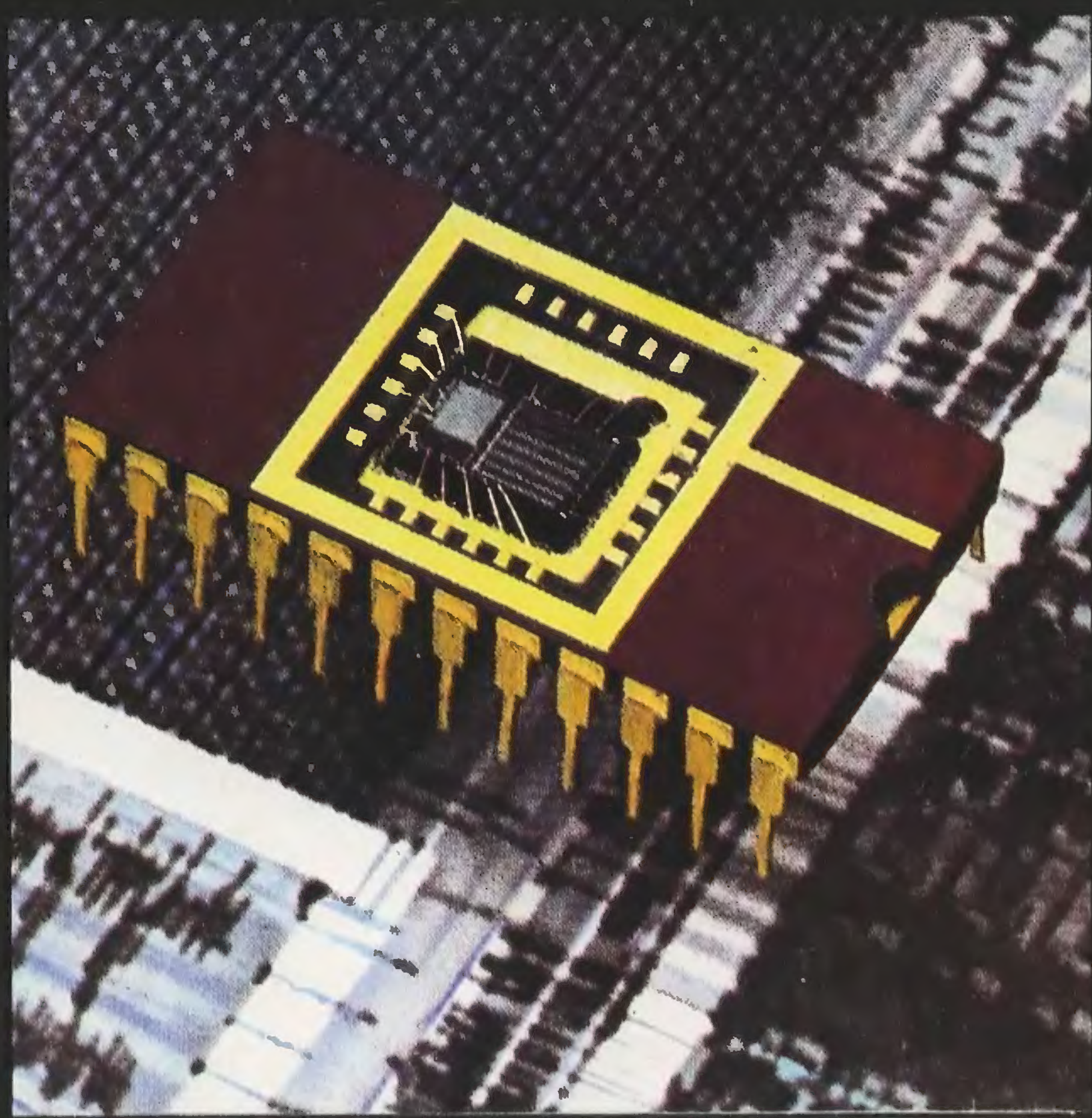
Подписная научно-популярная серия

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

И ЕЁ ПРИМЕНЕНИЕ

Остается загадкой "вопрос о том, каким образом удается построить столь совершенные архитектурные сооружения, имея под руками самый примитивный строительный материал... — триады кодовых знаков, которые миллиардами хранятся в ЭВМ на физических носителях".

Алан Кэй



Наш адрес:
СССР,
Москва,
Центр,
проезд
Серова, 4